```
DDDDDDDDDDDD        IIIIIIIII     RRRRRRRRRRRR
DDDDDDDDDDDD        IIIIIIIII     RRRRRRRRRRRR
DDDDDDDDDDDD        IIIIIIIII     RRRRRRRRRRRR
DDD         DDD        III        RRR         RRR
DDD         DDD        III        RRR         RRR
DDD         DDD        III        RRR         RRR
DDD         DDD        III        RRR         RRR
DDD         DDD        III        RRR         RRR
DDD         DDD        III        RRR         RRR
DDD         DDD        III        RRRRRRRRRRRR
DDD         DDD        III        RRRRRRRRRRRR
DDD         DDD        III        RRRRRRRRRRRR
DDD         DDD        III        RRR   RRR
DDD         DDD        III        RRR   RRR
DDD         DDD        III        RRR   RRR
DDD         DDD        III        RRR      RRR
DDD         DDD        III        RRR      RRR
DDD         DDD        III        RRR      RRR
DDDDDDDDDDDD        IIIIIIIII     RRR         RRR
DDDDDDDDDDDD        IIIIIIIII     RRR         RRR
DDDDDDDDDDDD        IIIIIIIII     RRR         RRR
```

```
**FILE**ID**DIRECTORY                                        D 14
```

```
DDDDDDDD      IIIIII    RRRRRRRR    EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT   000000    RRRRRRRR    YY      YY
DDDDDDDD      IIIIII    RRRRRRRR    EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT   000000    RRRRRRRR    YY      YY
DD       DD     II      RR      RR  EE          CC              TT     00       00 RR       RR YY      YY
DD       DD     II      RR      RR  EE          CC              TT     00       00 RR       RR YY      YY
DD       DD     II      RR      RR  EE          CC              TT     00       00 RR       RR   YY  YY
DD       DD     II      RR      RR  EE          CC              TT     00       00 RR       RR   YY  YY
DD       DD     II      RRRRRRRR    EEEEEEEE    CC              TT     00       00 RRRRRRRR        YY
DD       DD     II      RRRRRRRR    EEEEEEEE    CC              TT     00       00 RRRRRRRR        YY
DD       DD     II      RR  RR      EE          CC              TT     00       00 RR  RR          YY
DD       DD     II      RR  RR      EE          CC              TT     00       00 RR  RR          YY
DD       DD     II      RR    RR    EE          CC              TT     00       00 RR    RR        YY          ....
DD       DD     II      RR    RR    EE          CC              TT     00       00 RR    RR        YY          ....
DDDDDDDD      IIIIII    RR      RR  EEEEEEEEEE    CCCCCCCC      TT       000000    RR       RR     YY          ....
DDDDDDDD      IIIIII    RR      RR  EEEEEEEEEE    CCCCCCCC      TT       000000    RR       RR     YY          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE DIRECTORY (
0                  LANGUAGE (BLISS32),
0                  IDENT = 'V04-000',
0                  MAIN = DIR$MAIN
0                  ) =
0
1 BEGIN
1
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:      DIRECTORY
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the main processing routine for the directory
1 !       command.  It also contains various error reporting routines.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system, unprivileged user mode utilities.
1 !
1 !--
1 !
1 ! AUTHOR:        L. Mark Pilant          CREATION DATE:  3-Mar-1983
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-020 LMP0296         L. Mark Pilant,         6-Aug-1984  12:54
1 !               Note the hack to get /FULL to work with the magtape ACP.
1 !
1 !       V03-019 LMP0280         L. Mark Pilant,         19-Jul-1984  12:54
1 !               Give the correct text on the DIR$_SYNTAX error message.
1 !
1 !       V03-018 LMP0276         L. Mark Pilant,         11-Jul-1984  11:51
```

```
!               Some modifications:
!                 1)    Fix a bug in LMP0263 that caused extra headings to
!                       come out.
!                 2)    Fix the handling of /OUTPUT and /NOOUTPUT.
!
!       V03-017 LMP0263         L. Mark Pilant,         26-Jun-1984  12:58
!               Clear out the version count and saved directory name for
!               each input spec.
!
!       V03-016 JEJ0017         J E Johnson             16-Apr-1984
!               Fix bug caused by V03-014 edit.
!
!       V03-018 BLS0300         Benn Schreiber          11-APR-1984
!               Do not link with SECURESHR to get the format_acl service.
!               Rather, only load it if /acl or /full.
!
!       V03-014 JEJ0017         J E Johnson             27-Mar-1984
!               Clean up the network $SEARCH XAB fill support to use the
!               NOP flag SRCHXABS.
!
!       V03-013 LMP0211         L. Mark Pilant,         10-Mar-1984  12:44
!               Fix some minor logic problems that occurred when the display
!               logic was changed.
!
!       V03-012 BLS0265         Benn Schreiber          25-Jan-1984
!               Use enhanced lib$file_scan features for stickyness
!
!       V03-011 LMP0182         L. Mark Pilant,         11-Jan-1984  12:43
!               Note the use of the /SELECT qualifier with an appropriate flag.
!
!       V03-010 LMP0180         L. Mark Pilant,         12-Dec-1983  9:42
!               Correct a bug in the formatting uncovered by the fix in
!               LMP0176.
!
!       V03-009 LMP0176         L. Mark Pilant,         6-Dec-1983  8:54
!               Correct an incorrect piece of logic used to determine the
!               number of columns able to be printed in a display.
!
!       V03-008 LMP0171         L. Mark Pilant,         23-Nov-1983  10:39
!               Correct a bug that caused the size selection item to be
!               dropped on the floor.
!
!       V03-007 LMP0157         L. Mark Pilant,         27-Sep-1983  10:45
!               Add support for a unique message file.
!
!       V03-006 LMP0132         L. Mark Pilant,         3-Aug-1983  10:19
!               Correct the qualifier keyword COLUMN to be COLUMNS to match
!               the documentation.
!
!       V03-005 LMP0119         L. Mark Pilant,         15-Jun-1983  9:29
!               Add support for identifiers.
!
!       V03-004 LMP0108         L. Mark Pilant,         28-Apr-1983  10:49
!               Issue a DIRECTORY message if no files are found, not an RMS
!               message.  Also, add support for RMS journaling.
!
!       V03-003 LMP0100         L. Mark Pilant,         14-Apr-1983  11:49
```

```
1 !                     Misc fixups.
1 !
1 !             V03-002 LMP0096         L. Mark Pilant,         29-Mar-1983  10:01
1 !                     Correctly handle locked files.
1 !
1 !             V03-001 LMP0092         L. Mark Pilant,         25-Mar-1983  12:24
1 !                     Include the FHC XAB when /SIZE is specified.  Also fix
1 !                     the handling of the final error status.
1 !
1 !**
1
1 LIBRARY 'SYS$LIBRARY:LIB';
1 REQUIRE 'SRC$:DIRECTDEF';
```

```
!                       HACKS WORTH NOTING...
!
!       There are several hacks used by DIRECTORY to improve performance
!       and to compensate for bugs elsewhere in the system.
!
!       The first is mechanism that allows the file information requested
!       in the RMS XAB blocks to be filled in while performing a $SEARCH
!       over the network.  If the NAM block attached to the FAB doing the
!       $SEARCH has the NOP bit NAM$V_SRCHXABS set, then any XABs
!       attached to the FAB will have the requested information filled in
!       if it is available.
!
!       The next is used by LIB$FILE_SCAN to improve performance.  Doing
!       a $SEARCH operation over the network involves a considerable
!       ammount of startup overhead (to make the connection).  Therefore,
!       LIB$FILE_SCAN will only do the network $SEARCH operation if there
!       are wildcard characters present (as determined by the previous
!       $PARSE).  This means that if there are XABs to be filled, and no
!       wildcards are present in the filespec, it is necessary to issue
!       an explicit $SEARCH (outside of LIB$FILE_SCAN).
!
!       Another hack used here is to not explicitly link with SECURESHR,
!       which contains the format_acl service.  Rather, we auto-load it
!       using lib$find_image_symbol only if /acl or /full is present.  This
!       gives a reduction in activation time in the case we don't need
!       to format any acls.
!
!       The last hack is to make /FULL work with the magtape ACP.  There is
!       a bug in the magtape ACP encountered when doing wildcarding and
!       accessing by file name to the same tape drive.  The access by name
!       causes the magtape ACP to loose the wildcard context, resulting in
!       an infinite loop.  This is corrected in DIRECTORY by accessing the
!       file by "file-ID" even when /FULL is specified, if the device is a
!       sequential device.
```

```
1 FORWARD ROUTINE
1         DIR$MAIN,                               ! Main processing routine
1         DIR$GET_FILE,                           ! Get a file spec to process
1         DIR$INPUT_ERROR,                        ! Signal file scanning error
1         DIR$FILE_ERROR,                         ! Signal file error
1         DIR$OUTPUT;                             ! General output routine
1
1 OWN
1         FORMAT_ACL_ADDR,                        ! Address of real SYS$FORMAT_ACL
1         OUTPUT_FAB      : $FAB_DECL,            ! Output file RMS structures
1
1 ! OUTPUT_RAB is in DIRECTDEF.REQ because it is referenced by the SIGNAL macro
1 ! to flush out the RMS buffers when an error occurs.
1
1         OUTPUT_NAM      : $NAM_DECL,
1         OUT_EXP_NAM     : $BBLOCK [NAM$C_MAXRSS],
1         OUT_RES_NAM     : $BBLOCK [NAM$C_MAXRSS];
1
1 EXTERNAL ROUTINE
1         CLI$GET_VALUE   : ADDRESSING_MODE (GENERAL),     ! Get a qualifier value
1         CLI$PRESENT     : ADDRESSING_MODE (GENERAL),     ! See if qualifier present
1         LIB$FILE_SCAN   : ADDRESSING_MODE (GENERAL),     ! Search wildcard file spec
1         LIB$FIND_IMAGE_SYMBOL : ADDRESSING_MODE(GENERAL),! Image activate
1
1 ! Following are the common qualifier scanning routines
1
1         LIB$QUAL_FILE_PARSE      : ADDRESSING_MODE (GENERAL);    ! Set up select
```

```
1 ROUTINE DIR$MAIN =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This routine is the main processing routine for the DIRECTORY command.
1 !         It parses the qualifiers in the command line to determine what
1 !         information is to be displayed for the selected file or files.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !         DIR$MAIN ()
1 !
1 ! INPUT PARAMETERS:
1 !         none
1 !
1 ! IMPLICIT INPUTS:
1 !         none
1 !
1 ! OUTPUT PARAMETERS:
1 !         none
1 !
1 ! IMPLICIT OUTPUTS:
1 !         none
1 !
1 ! ROUTINE VALUE:
1 !         The worst error encountered or SS$_NORMAL.
1 !
1 ! SIDE EFFECTS:
1 !         none
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2         STATUS,                                         ! Local routine exit status
2         CLI_STATUS,                                     ! CLI parse status
2         SCAN_CONTEXT,                                   ! filescan context
2         INPUT_FAB       : $FAB_DECL,                    ! Input file RMS structures
2         INPUT_NAM       : $NAM_DECL,
2         INP_EXP_NAM     : $BBLOCK [NAM$C_MAXRSS],
2         INP_RES_NAM     : $BBLOCK [NAM$C_MAXRSS],
2         FILE_DESC       : $BBLOCK [DSC$C_S_BLN],              ! File name descr
2         VALUE_DESC      : $BBLOCK [DSC$C_S_BLN],              ! Qualifier value
2         GETDVI_ARGS     : VECTOR [7],                   ! GETDVI argument list
2         INDEV_CLASS,                                    ! Input device class
2         INDEV_BUFSIZ,                                   ! Input device buffer size
2         XAB_PTR         : REF $BBLOCK;                  ! Pointer to current XAB
2
2 EXTERNAL LITERAL
2         CLI$_DEFAULTED,                                 ! Value present by default
2         CLI$_NEGATED;                                   ! Qualifier negated
2
2 EXTERNAL ROUTINE
2         DIR$GET_INFO,                                   ! Get information about a file
```

```
        DIR$TOTAL,                                  ! Type out per directory totals
        DIR$GRAND_TOTAL,                            ! Type out grand total info
        LIB$CVT_DTB     : ADDRESSING_MODE (GENERAL),    ! Convert string to value
        LIB$GET_VM      : ADDRESSING_MODE (GENERAL);    ! Allocate dynamic memory

! DIRECTORY error messages

EXTERNAL LITERAL
        DIR$_NOFILES;

! Initialize all variables

SCAN_CONTEXT = 0;
QUAL_FLAGS = 0;
WORST_ERROR = SS$_NORMAL;
CHANNEL = 0;
CH$FILL (0, NAM$C_DVI, DEVICE_NAME);
COLUMN_COUNT = COLUMN_INDEX = COLUMN_WIDTH = 0;
VERSION_COUNT = VERSION_INDEX = 0;
PREV_DIR_LEN = PREV_FILE_LEN = 0;
TOTAL_USED = TOTAL_ALLOC = TOTAL_FILES = 0;
GRAND_USED = GRAND_ALLOC = GRAND_FILES = GRAND_DIRS = 0;
COLUMN_WIDTH = 0;
INDEV_CLASS = INDEV_BUFSIZ = 0;
FIRST_XAB = XAB_PTR = 0;
CH$FILL (0, DSC$C_S_BLN, VALUE_DESC);
VALUE_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
CH$MOVE (DSC$C_S_BLN, VALUE_DESC, FILE_DESC);
CH$MOVE (DSC$C_S_BLN, VALUE_DESC, LINE_DESC);
LINE_DESC[DSC$A_POINTER] = LINE_BUFFER;

! Get the block of memory needed to hold the display information.

STATUS = LIB$GET_VM (%REF (DIR_C_LENGTH), DISPLAY_BLOCK);
IF NOT .STATUS
THEN
    BEGIN
    SIGNAL (.STATUS);
    RETURN .WORST_ERROR;
    END;

! Initialize all RMS data structures.

$FAB_INIT       (FAB = INPUT_FAB,               ! Init input structures
                 DNA = UPLIT ('*.*;*'),
                 DNS = %CHARCOUNT ('*.*;*'),
                 NAM = INPUT_NAM);
$NAM_INIT       (NAM = INPUT_NAM,
                 ESA = INP_EXP_NAM,
                 ESS = NAM$C_MAXRSS,
                 RSA = INP_RES_NAM,
                 RSS = NAM$C_MAXRSS);

$FAB_INIT       (FAB = OUTPUT_FAB,              ! Init output structures
                 DNA = UPLIT ('DIRECTORY.LIS'),
                 DNS = %CHARCOUNT ('DIRECTORY.LIS'),
                 FAC = PUT,
P 0705 2                      FOP = SQO,
P 0706 2                      NAM = OUTPUT_NAM,
  0707 2                      RAT = CR);
P 0708 2 $RAB_INIT           (RAB = OUTPUT_RAB,
  0709 2                      FAB = OUTPUT_FAB);
P 0710 2 $NAM_INIT           (NAM = OUTPUT_NAM,
P 0711 2                      ESA = OUT_EXP_NAM,
P 0712 2                      ESS = NAM$C_MAXRSS,
P 0713 2                      RSA = OUT_RES_NAM,
  0714 2                      RSS = NAM$C_MAXRSS);
  0715 2
  0716 2 ! Parse the various command qualifiers that may have been given on the
  0717 2 ! command line.
  0718 2
  0719 2 ! First check for any of the common qualifiers to determine what XABs
  0720 2 ! may be needed.
  0721 2
  0722 2 IF CLI$PRESENT ($DESCRIPTOR ('BEFORE'))
  0723 2 OR CLI$PRESENT ($DESCRIPTOR ('SINCE'))
  0724 2 THEN
  0725 3     BEGIN
  0726 3     QUAL_FLAGS[DIR_V_NEED_DAT] = 1;              ! DAT XAB required
  0727 3     QUAL_FLAGS[DIR_V_COMM_QUAL] = 1;
  0728 2     END;
  0729 2
  0730 2 IF CLI$PRESENT ($DESCRIPTOR ('BY_OWNER'))
  0731 2 THEN
  0732 3     BEGIN
  0733 3     QUAL_FLAGS[DIR_V_NEED_PRO] = 1;              ! PRO XAB required
  0734 3     QUAL_FLAGS[DIR_V_COMM_QUAL] = 1;
  0735 2     END;
  0736 2
  0737 2 ! Now check for all the display tayloring qualifiers
  0738 2
  0739 2 QUAL_FLAGS[DIR_V_QUAL_ACL] = CLI$PRESENT ($DESCRIPTOR ('ACL'));
  0740 2 QUAL_FLAGS[DIR_V_QUAL_BRIE] = CLI$PRESENT ($DESCRIPTOR ('BRIEF'));
  0741 3 IF (CLI_STATUS = QUAL_FLAGS[DIR_V_QUAL_COLU] = CLI$PRESENT ($DESCRIPTOR ('COLUMNS')))
  0742 2 THEN
  0743 3     BEGIN
  0744 3     CLI$GET_VALUE ($DESCRIPTOR ('COLUMNS'), VALUE_DESC);
  0745 3     STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
  0746 3                           .VALUE_DESC[DSC$A_POINTER],
  0747 3                           COLUMN_COUNT);
  0748 3     IF NOT .STATUS OR .COLUMN_COUNT LSS 0
  0749 3     THEN
  0750 4         BEGIN
  0751 4         SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
  0752 4         RETURN .WORST_ERROR;
  0753 3         END;
  0754 3     IF .COLUMN_COUNT EQL 0 THEN COLUMN_COUNT = 1;
  0755 3     IF .CLI_STATUS EQL CLI$_DEFAULTED THEN QUAL_FLAGS[DIR_V_COLU_DEF] = 1;
  0756 2     END;
  0757 3 IF (QUAL_FLAGS[DIR_V_QUAL_DATE] = CLI$PRESENT ($DESCRIPTOR ('DATE')))
  0758 2 THEN
  0759 3     BEGIN
  0760 3     QUAL_FLAGS[DIR_V_NEED_DAT] = 1;              ! DAT XAB required
  0761 3     IF CLI$PRESENT ($DESCRIPTOR ('DATE.ALL'))
```

```
3           THEN
4               BEGIN
4               QUAL_FLAGS[DIR_V_DATE_CRE] = 1;
4               QUAL_FLAGS[DIR_V_DATE_EXP] = 1;
4               QUAL_FLAGS[DIR_V_DATE_MOD] = 1;
4               QUAL_FLAGS[DIR_V_DATE_BAK] = 1;
4               COLUMN_WIDTH = .COLUMN_WIDTH + 19 * 4;
4               END
3           ELSE
4               BEGIN
4               IF CLI$PRESENT ($DESCRIPTOR ('DATE.CREATED'))
4               THEN
5                   BEGIN
5                   QUAL_FLAGS[DIR_V_DATE_CRE] = 1;
5                   COLUMN_WIDTH = .COLUMN_WIDTH + 19;
4                   END;
4               IF CLI$PRESENT ($DESCRIPTOR ('DATE.EXPIRED'))
4               THEN
5                   BEGIN
5                   QUAL_FLAGS[DIR_V_DATE_EXP] = 1;
5                   COLUMN_WIDTH = .COLUMN_WIDTH + 19;
4                   END;
4               IF CLI$PRESENT ($DESCRIPTOR ('DATE.MODIFIED'))
4               THEN
5                   BEGIN
5                   QUAL_FLAGS[DIR_V_DATE_MOD] = 1;
5                   COLUMN_WIDTH = .COLUMN_WIDTH + 19;
4                   END;
4               IF CLI$PRESENT ($DESCRIPTOR ('DATE.BACKUP'))
4               THEN
5                   BEGIN
5                   QUAL_FLAGS[DIR_V_DATE_BAK] = 1;
5                   COLUMN_WIDTH = .COLUMN_WIDTH + 19;
4                   END;
3               END;
2           END;
3       IF (QUAL_FLAGS[DIR_V_QUAL_FID] = CLI$PRESENT ($DESCRIPTOR ('FILE_ID')))
2       THEN COLUMN_WIDTH = .COLUMN_WIDTH + 21;
3       IF (QUAL_FLAGS[DIR_V_QUAL_FULL] = CLI$PRESENT ($DESCRIPTOR ('FULL')))
2       THEN
3           BEGIN
3           QUAL_FLAGS[DIR_V_NEED_FHC] = QUAL_FLAGS[DIR_V_NEED_DAT] = 1;
3           QUAL_FLAGS[DIR_V_NEED_PRO] = QUAL_FLAGS[DIR_V_NEED_SUM] = 1;
3           QUAL_FLAGS[DIR_V_NEED_JNL] = 1;
3           END;
2       QUAL_FLAGS[DIR_V_QUAL_GRAN] = CLI$PRESENT ($DESCRIPTOR ('GRAND_TOTAL'));
2       QUAL_FLAGS[DIR_V_QUAL_HEAD] = CLI$PRESENT ($DESCRIPTOR ('HEADING'));
2
2       ! /PRINTER is checked out of sequence because it may affect how /OUTPUT is
2       ! handled.
2
3       IF (QUAL_FLAGS[DIR_V_QUAL_PRIN] = CLI$PRESENT ($DESCRIPTOR ('PRINTER')))
2       THEN
3           BEGIN
3           OUTPUT_FAB[FAB$V_SPL] = 1;                  ! Spool file when closed.
3           OUTPUT_FAB[FAB$V_DLT] = 1;                  ! Delete file after printing
2           END;
```

```
3 IF (CLI_STATUS = QUAL_FLAGS[DIR_V_QUAL_OUTP] = CLI$PRESENT ($DESCRIPTOR ('OUTPUT')))
2 THEN
3     BEGIN
3     CLI$GET_VALUE ($DESCRIPTOR ('OUTPUT'), FILE_DESC);
3     OUTPUT_FAB[FAB$L_FNA] = .FILE_DESC[DSC$A_POINTER];
3     IF (OUTPUT_FAB[FAB$B_FNS] = .FILE_DESC[DSC$W_LENGTH]) EQL 0
3     AND NOT .QUAL_FLAGS[DIR_V_QUAL_PRTN]
3     THEN
4         BEGIN
4         OUTPUT_FAB[FAB$L_FNA] = UPLIT ('SYS$OUTPUT:');
4         OUTPUT_FAB[FAB$B_FNS] = %CHARCOUNT ('SYS$OUTPUT:');
3         END;
3     END
2 ELSE
3     BEGIN
3     IF .CLI_STATUS EQL CLI$_NEGATED
3     THEN
4         BEGIN
4         OUTPUT_FAB[FAB$L_FNA] = UPLIT ('NL:');
4         OUTPUT_FAB[FAB$B_FNS] = %CHARCOUNT ('NL:');
4         OUTPUT_FAB[FAB$V_SPL] = 0;
4         OUTPUT_FAB[FAB$V_DLT] = 0;
3         END;
2     END;
3 IF (QUAL_FLAGS[DIR_V_QUAL_OWNE] = CLI$PRESENT ($DESCRIPTOR ('OWNER')))
2 THEN
3     BEGIN
3     QUAL_FLAGS[DIR_V_NEED_PRO] = 1;
3     QUAL_FLAGS[DIR_V_USE_ID] = CLI$PRESENT ($DESCRIPTOR ('OWNER.IDENTIFIER'));
2     END;
3 IF (QUAL_FLAGS[DIR_V_QUAL_PROT] = CLI$PRESENT ($DESCRIPTOR ('PROTECTION')))
2 THEN
3     BEGIN
3     QUAL_FLAGS[DIR_V_NEED_PRO] = 1;
3     COLUMN_WIDTH = .COLUMN_WIDTH + 23;
2     END;
3 IF (QUAL_FLAGS[DIR_V_QUAL_SECU] = CLI$PRESENT ($DESCRIPTOR ('SECURITY')))
2 THEN
3     BEGIN
3     QUAL_FLAGS[DIR_V_NEED_PRO] = 1;
3     QUAL_FLAGS[DIR_V_QUAL_ACL] = QUAL_FLAGS[DIR_V_QUAL_OWNE] =
3                                  QUAL_FLAGS[DIR_V_QUAL_PROT] = 1;
3     COLUMN_WIDTH = .COLUMN_WIDTH + 23;
2     END;
2 IF CLI$PRESENT ($DESCRIPTOR ('SELECT'))
2 THEN
3     BEGIN
3     MIN_BLOCK = 0;                                   !*****
3     MAX_BLOCK = 1073741823;                          !*****
3     IF CLI$PRESENT ($DESCRIPTOR ('SELECT.SIZE.MINIMUM_SIZE'))
3     THEN
4         BEGIN
4         QUAL_FLAGS[DIR_V_SELE_SIZE] = 1;
4         CLI$GET_VALUE ($DESCRIPTOR ('SELECT.SIZE.MINIMUM_SIZE'), VALUE_DESC);
4         STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
4                               .VALUE_DESC[DSC$A_POINTER],
4                               MIN_BLOCK);
```

```
4               IF NOT .STATUS OR .MIN_BLOCK LSS 0
4               THEN
5                   BEGIN
5                   SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
5                   RETURN .WORST_ERROR;
4                   END;
4               QUAL_FLAGS[DIR_V_NEED_FHC] = 1;
3               END;
3           IF CLI$PRESENT ($DESCRIPTOR ('SELECT.SIZE.MAXIMUM_SIZE'))
3           THEN
4               BEGIN
4               QUAL_FLAGS[DIR_V_SELE_SIZE] = 1;
4               CLI$GET_VALUE ($DESCRIPTOR ('SELECT.SIZE.MAXIMUM_SIZE'), VALUE_DESC);
4               STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
4                                     .VALUE_DESC[DSC$A_POINTER],
4                                     MAX_BLOCK);
4               IF NOT .STATUS OR .MAX_BLOCK LSS 0
4               THEN
5                   BEGIN
5                   SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
5                   RETURN .WORST_ERROR;
4                   END;
4               QUAL_FLAGS[DIR_V_NEED_FHC] = 1;
3               END;
2           END;
3       IF (QUAL_FLAGS[DIR_V_QUAL_SIZE] = CLI$PRESENT ($DESCRIPTOR ('SIZE')))
2       THEN
3           BEGIN
3           QUAL_FLAGS[DIR_V_NEED_FHC] = 1;
3           IF CLI$PRESENT ($DESCRIPTOR ('SIZE.ALL'))
3           THEN QUAL_FLAGS[DIR_V_SIZE_ALL] = 1;
3           IF CLI$PRESENT ($DESCRIPTOR ('SIZE.ALLOCATION'))
3           THEN QUAL_FLAGS[DIR_V_SIZE_ALLO] = 1;
3           IF CLI$PRESENT ($DESCRIPTOR ('SIZE.USED'))
3           THEN QUAL_FLAGS[DIR_V_SIZE_USED] = 1;
2           END;
2       QUAL_FLAGS[DIR_V_QUAL_TOTL] = CLI$PRESENT ($DESCRIPTOR ('TOTAL'));
2       QUAL_FLAGS[DIR_V_QUAL_TRAI] = CLI$PRESENT ($DESCRIPTOR ('TRAILING'));
3       IF (QUAL_FLAGS[DIR_V_QUAL_VERS] = CLI$PRESENT ($DESCRIPTOR ('VERSIONS')))
2       THEN
3           BEGIN
3           CLI$GET_VALUE ($DESCRIPTOR ('VERSIONS'), VALUE_DESC);
3           STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
3                                 .VALUE_DESC[DSC$A_POINTER],
3                                 VERSION_COUNT);
3           IF NOT .STATUS OR .VERSION_COUNT LEQ 0
3           THEN
4               BEGIN
4               SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
4               RETURN .WORST_ERROR;
3               END;
2           END;
3       IF (QUAL_FLAGS[DIR_V_QUAL_WIDT] = CLI$PRESENT ($DESCRIPTOR ('WIDTH')))
2       THEN
3           BEGIN
3           CLI$GET_VALUE ($DESCRIPTOR ('WIDTH.DISPLAY'), VALUE_DESC);
3           STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
                                  .VALUE_DESC[DSC$A_POINTER],
                                  DISPLAY_WIDTH);
        IF NOT .STATUS OR .DISPLAY_WIDTH LSS 0                  !*****
        THEN
            BEGIN
            SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
            RETURN .WORST_ERROR;
            END;
        CLI$GET_VALUE ($DESCRIPTOR ('WIDTH.FILENAME'), VALUE_DESC);
        STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
                              .VALUE_DESC[DSC$A_POINTER],
                              FILENAME_WIDTH);
        IF NOT .STATUS OR .FILENAME_WIDTH LSS 0                 !*****
        THEN
            BEGIN
            SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
            RETURN .WORST_ERROR;
            END;
        IF .FILENAME_WIDTH EQL 0 THEN FILENAME_WIDTH = 19;      !*****
        CLI$GET_VALUE ($DESCRIPTOR ('WIDTH.OWNER'), VALUE_DESC);
        STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
                              .VALUE_DESC[DSC$A_POINTER],
                              OWNER_WIDTH);
        IF NOT .STATUS OR .OWNER_WIDTH LSS 0                    !*****
        THEN
            BEGIN
            SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
            RETURN .WORST_ERROR;
            END;
        IF .OWNER_WIDTH EQL 0 THEN OWNER_WIDTH = 20;            !*****
        CLI$GET_VALUE ($DESCRIPTOR ('WIDTH.SIZE'), VALUE_DESC);
        STATUS = LIB$CVT_DTB (.VALUE_DESC[DSC$W_LENGTH],
                              .VALUE_DESC[DSC$A_POINTER],
                              SIZE_WIDTH);
        IF NOT .STATUS OR .SIZE_WIDTH LSS 0                     !*****
        THEN
            BEGIN
            SIGNAL (DIR$_SYNTAX, 1, VALUE_DESC);
            RETURN .WORST_ERROR;
            END;
        IF .SIZE_WIDTH EQL 0 THEN SIZE_WIDTH = 6;               !*****
        END;

    ! Open the specified output file/device.

    STATUS = $CREATE (FAB = OUTPUT_FAB);
    IF NOT .STATUS
    THEN
        BEGIN
        DIR$FILE_ERROR (DIR$_OPENOUT, OUTPUT_FAB);
        RETURN .WORST_ERROR;
        END;
    STATUS = $CONNECT (RAB = OUTPUT_RAB);
    IF NOT .STATUS
    THEN
        BEGIN
        DIR$FILE_ERROR (DIR$_OPENOUT, OUTPUT_FAB);
```

```
  3         RETURN .WORST_ERROR;
  2         END;
  2
  2     ! Determine the width of the output device.
  2
  2     IF .(OUTPUT_FAB[FAB$L_DEV])<$BITPOSITION (DEV$V_TRM), 1>
  2     THEN
  3         BEGIN
  3         CH$FILL (0, 7*4, GETDVI_ARGS);
  3         GETDVI_ARGS[0] = DVI$_DEVCLASS^16 OR 4;
  3         GETDVI_ARGS[1] = INDEV_CLASS;
  3         GETDVI_ARGS[3] = DVI$_DEVBUFSIZ^16 OR 4;
  3         GETDVI_ARGS[4] = INDEV_BUFSIZ;
  3
P 3         STATUS = $GETDVI (DEVNAM = $DESCRIPTOR ('SYS$OUTPUT'),
  3                           ITMLST = GETDVI_ARGS);
  3         IF NOT .STATUS
  3         THEN
  4             BEGIN
  4             SIGNAL (.STATUS);
  4             RETURN .WORST_ERROR;
  3             END;
  2         END;
  2     IF .DISPLAY_WIDTH EQL 0
  2     THEN
  3         BEGIN
  3         IF .INDEV_CLASS NEQ DC$_TERM THEN INDEV_BUFSIZ = 132;
  3         DISPLAY_WIDTH = .INDEV_BUFSIZ;
  2         END;
  2
  2     ! If the number of columns is defaulted and an information qualifier is
  2     ! specified, set the column count to 1.
  2
  3     IF (.QUAL_FLAGS[DIR_V_QUAL_DATE] OR .QUAL_FLAGS[DIR_V_QUAL_OWNE]
  3         OR .QUAL_FLAGS[DIR_V_QUAL_PROT] OR .QUAL_FLAGS[DIR_V_QUAL_SIZE]
  3         OR .QUAL_FLAGS[DIR_V_QUAL_FID] OR NOT .QUAL_FLAGS[DIR_V_QUAL_HEAD])
  2     AND .QUAL_FLAGS[DIR_V_COLU_DEF]
  2     THEN COLUMN_COUNT = 1;
  2
  2     ! Check to see if XABs are needed to gather information.
  2
  2     IF .QUAL_FLAGS[DIR_V_NEED_FHC]
  2     THEN
  3         BEGIN
  3         IF .FIRST_XAB EQL 0
  3         THEN FIRST_XAB = XAB_PTR = INFO_XABFHC
  3         ELSE (XAB_PTR[XAB$L_NXT] = INFO_XABFHC; XAB_PTR = INFO_XABFHC);
  2         END;
  2     IF .QUAL_FLAGS[DIR_V_NEED_DAT]
  2     THEN
  3         BEGIN
  3         IF .FIRST_XAB EQL 0
  3         THEN FIRST_XAB = XAB_PTR = INFO_XABDAT
  3         ELSE (XAB_PTR[XAB$L_NXT] = INFO_XABDAT; XAB_PTR = INFO_XABDAT);
  2         END;
  2     IF .QUAL_FLAGS[DIR_V_NEED_PRO]
  2     THEN
```

```
    BEGIN
    IF .FIRST_XAB EQL 0
    THEN FIRST_XAB = XAB_PTR = INFO_XABPRO
    ELSE (XAB_PTR[XAB$L_NXT] = INFO_XABPRO; XAB_PTR = INFO_XABPRO);
    END;
IF .QUAL_FLAGS[DIR_V_NEED_SUM]
THEN
    BEGIN
    IF .FIRST_XAB EQL 0
    THEN FIRST_XAB = XAB_PTR = INFO_XABSUM
    ELSE (XAB_PTR[XAB$L_NXT] = INFO_XABSUM; XAB_PTR = INFO_XABSUM);
    END;
IF .QUAL_FLAGS[DIR_V_NEED_JNL]
THEN
    BEGIN
    IF .FIRST_XAB EQL 0
    THEN FIRST_XAB = XAB_PTR = INFO_XABJNL
    ELSE (XAB_PTR[XAB$L_NXT] = INFO_XABJNL; XAB_PTR = INFO_XABJNL);
    INFO_XABJNL[XAB$L_AIA] = DISPLAY_BLOCK[DIR_T_AI_NAME];
    INFO_XABJNL[XAB$B_AIS] = XAB$C_MAXJNLNAM;
    INFO_XABJNL[XAB$L_BIA] = DISPLAY_BLOCK[DIR_T_BI_NAME];
    INFO_XABJNL[XAB$B_BIS] = XAB$C_MAXJNLNAM;
    INFO_XABJNL[XAB$L_ATA] = DISPLAY_BLOCK[DIR_T_AT_NAME];
    INFO_XABJNL[XAB$B_ATS] = XAB$C_MAXJNLNAM;
    END;

! At this point all of the qualifiers have been parsed.  Now determine the
! column width and the maximum number of columns that can be printed given
! specified (or default) display width.  This value is minimized with the
! value given on the /COLUMN qualifier.

COLUMN_WIDTH = .COLUMN_WIDTH + .FILENAME_WIDTH + 1;
IF .QUAL_FLAGS[DIR_V_QUAL_OWNE] THEN COLUMN_WIDTH = .COLUMN_WIDTH + .OWNER_WIDTH + 2;
IF .QUAL_FLAGS[DIR_V_QUAL_SIZE]
THEN
    BEGIN
    IF .QUAL_FLAGS[DIR_V_SIZE_ALL]
    THEN COLUMN_WIDTH = .COLUMN_WIDTH + .SIZE_WIDTH * 2 + 2
    ELSE COLUMN_WIDTH = .COLUMN_WIDTH + .SIZE_WIDTH + 2;
    END;
IF (.QUAL_FLAGS[DIR_V_DATE_CRE] OR .QUAL_FLAGS[DIR_V_DATE_MOD]
    OR .QUAL_FLAGS[DIR_V_DATE_EXP] OR .QUAL_FLAGS[DIR_V_DATE_BAK]
    OR .QUAL_FLAGS[DIR_V_QUAL_OWNE] OR .QUAL_FLAGS[DIR_V_QUAL_PROT]
    OR .QUAL_FLAGS[DIR_V_QUAL_SIZE] OR .QUAL_FLAGS[DIR_V_QUAL_FID])
THEN
    BEGIN
    COLUMN_WIDTH = .COLUMN_WIDTH + 4;
    COLUMN_COUNT = MINU (.COLUMN_COUNT, (.DISPLAY_WIDTH + 4) / .COLUMN_WIDTH);
    END
ELSE COLUMN_COUNT = MINU (.COLUMN_COUNT, .DISPLAY_WIDTH / .COLUMN_WIDTH);
IF .COLUMN_COUNT LEQ 0 OR .QUAL_FLAGS[DIR_V_QUAL_ACL] THEN COLUMN_COUNT = 1;

! LIB$QUAL_FILE_PARSE is going to parse the common qualifiers.  It sets up
! a data base which describes the results for LIB$QUAL_FILE_MATCH to use.

STATUS = LIB$QUAL_FILE_PARSE (%REF (LIB$M_CQF_BACKUP OR
                                    LIB$M_CQF_BEFORE OR
                                    LIB$M_CQF_CREATED OR
                                    LIB$M_CQF_EXCLUDE OR
                                    LIB$M_CQF_EXPIRED OR
                                    LIB$M_CQF_MODIFIED OR
                                    LIB$M_CQF_SINCE OR
                                    LIB$M_CQF_BYOWNER
                                    ), CMR_QUAL_CTX);
IF NOT .STATUS
THEN
    BEGIN
    SIGNAL (.STATUS);
    RETURN .WORST_ERROR;
    END;

CLI$GET_VALUE ($DESCRIPTOR ('INPUT'), FILE_DESC);
INPUT_FAB[FAB$L_FNA] = .FILE_DESC[DSC$A_POINTER];
INPUT_FAB[FAB$B_FNS] = .FILE_DESC[DSC$W_LENGTH];

!
! If /FULL or /ACL, then image activate SECURESHR, which contains
! the routine SYS$FORMAT_ACL.
!
IF .QUAL_FLAGS[DIR_V_QUAL_FULL]
    OR .QUAL_FLAGS[DIR_V_QUAL_ACL]
THEN BEGIN
    STATUS = LIB$FIND_IMAGE_SYMBOL($DESCRIPTOR('SECURESHR'),
                                $DESCRIPTOR('SYS$FORMAT_ACL'),FORMAT_ACL_ADDR);
    IF NOT .STATUS
    THEN BEGIN
        SIGNAL (.STATUS);
        RETURN .WORST_ERROR;
        END;
    END;

! Process each file specification specified in the command line.

DO
    BEGIN

! The following is a KLUDGE to get the XAB information across the network.
! If the NOP field of the NAM block has the SRCHXABS flag set, then any
! XABs (supported by the DAP protocol) connected to the FAB are filled in.

    IF .QUAL_FLAGS[DIR_V_NEED_FHC] OR .QUAL_FLAGS[DIR_V_NEED_DAT]
    OR .QUAL_FLAGS[DIR_V_NEED_PRO] OR .QUAL_FLAGS[DIR_V_NEED_SUM]
    OR .QUAL_FLAGS[DIR_V_NEED_JNL]
    THEN
        BEGIN
        INPUT_NAM[NAM$V_SRCHXABS] = 1;
        INPUT_FAB[FAB$L_XAB] = .FIRST_XAB;
        END;

    LIB$FILE_SCAN (INPUT_FAB,
                   DIR$GET_INFO,                ! File found action routine
                   DIR$INPUT_ERROR,             ! Input error action routine
                   SCAN_CONTEXT);               ! Context for stickyness
    END
```

```
UNTIL NOT DIR$GET_FILE(INPUT_FAB);

IF .LINE_DESC[DSC$W_LENGTH] GTR 0 THEN DIR$OUTPUT (0, LINE_DESC);
IF .TOTAL_FILES NEQ 0 THEN DIR$TOTAL ();
IF .GRAND_DIRS GTR 1
OR .QUAL_FLAGS[DIR_V_QUAL_GRAN]
THEN DIR$GRAND_TOTAL ();                          ! Display grand totals

! If no files have been selected, and no other errors have occurred, return
! a status of RMS$_FNF instead of success.

IF .WORST_ERROR AND NOT .QUAL_FLAGS[DIR_V_FILE_FOUND]
THEN
    BEGIN
    SIGNAL (DIR$_NOFILES);
    WORST_ERROR = (RMS$_FNF AND NOT STS$M_SEVERITY) OR STS$K_WARNING
                                                   OR STS$M_INHIB_MSG;
    END;

STATUS = $CLOSE (FAB = OUTPUT_FAB);
IF NOT .STATUS THEN DIR$FILE_ERROR (DIR$_CLOSEOUT, OUTPUT_FAB);

RETURN .WORST_ERROR;

END;                                              ! End of routine DIR_MAIN

                                                   .TITLE  DIRECTORY
                                                   .IDENT  \V04-000\

                                                   .PSECT  DIR$COMMON,NOEXE,  OVR,0

                                   00000 QUAL_FLAGS:
                                                  .BLKB    8
                                   00008 COLUMN_COUNT:
                                                  .BLKB    4
                                   0000C COLUMN_INDEX:
                                                  .BLKB    4
                                   00010 COLUMN_WIDTH:
                                                  .BLKB    4
                                   00014 WORST_ERROR:
                                                  .BLKB    4
                                   00018 CMN_QUAL_CTX:
                                                  .BLKB    4
                                   0001C DISPLAY_BLOCK:
                                                  .BLKB    4
                                   00020 CHANNEL:.BLKB    4
                                   00024 DEVICE_NAME:
                                                  .BLKB    16
                                   00034 LINE_DESC:
                                                  .BLKB    8
                                   0003C LINE_BUFFER:
                                                  .BLKB    1024
                                   0043C TOTAL_USED:
                                                  .BLKB    4
                                   00440 TOTAL_ALLOC:
                                                  .BLKB    4
```

```
                00444 TOTAL_FILES:
                                .BLKB   4
                00448 GRAND_USED:
                                .BLKB   4
                0044C GRAND_ALLOC:
                                .BLKB   4
                00450 GRAND_FILES:
                                .BLKB   4
                00454 GRAND_DIRS:
                                .BLKB   4
                00458 PREV_DIR:
                                .BLKB   255
                00557           .BLKB   1
                00558 PREV_DIR_LEN:
                                .BLKB   4
                0055C PREV_FILE:
                                .BLKB   255
                0065B           .BLKB   1
                0065C PREV_FILE_LEN:
                                .BLKB   4
                00660 VERSION_COUNT:
                                .BLKB   4
                00664 VERSION_INDEX:
                                .BLKB   4
                00668 FIRST_XAB:
                                .BLKB   4
          22    0066C INFO_XABJNL:
                                .BYTE   34
          3C    0066D           .BYTE   60
        0000    0066E           .WORD   0
    00000000    00670           .LONG   0
        0000    00674           .WORD   0
        0000    00676           .WORD   0
          00    00678           .BYTE   0
          00    00679           .BYTE   0
        0000    0067A           .WORD   0
    00000000    0067C           .LONG   0
          00    00680           .BYTE   0
          00    00681           .BYTE   0
        0000    00682           .WORD   0
    00000000    00684           .LONG   0
          00    00688           .BYTE   0
          00    00689           .BYTE   0
        0000    0068A           .WORD   0
    00000000    0068C           .LONG   0
                00690           .BLKB   24
          16    006A8 INFO_XABSUM:
                                .BYTE   22
          0C    006A9           .BYTE   12
        0000    006AA           .WORD   0
    00000000    006AC           .LONG   0
          00    006B0           .BYTE   0
          00    006B1           .BYTE   0
        0000    006B2           .WORD   0
          13    006B4 INFO_XABPRO:
                                .BYTE   19
          58    006B5           .BYTE   88
```

```
                0000  006B6                .WORD   0
            00000000  006B8                .LONG   0
                FFFF  006BC                .WORD   -1
                  00  006BE                .BYTE   0
                  00  006BF                .BYTE   0
          0000  0000  006C0                .WORD   0, 0
                  00  006C4                .BYTE   0
                  00  006C5                .BYTE   0
                0000  006C6                .WORD   0
            00000000  006C8                .LONG   0
            00000000  006CC                .LONG   0
                0000  006D0                .WORD   0
                0000  006D2                .WORD   0
            00000000  006D4                .LONG   0
            00000000  006D8                .LONG   0
                      006DC                .BLKB   48
                  12  0070C  INFO_XABDAT:
                                           .BYTE   18
                  2C  0070D                .BYTE   44
                0000  0070E                .WORD   0
            00000000  00710                .LONG   0
                0000  00714                .WORD   0
                0000  00716                .WORD   0
           00000000#  00718                .LONG   0[2]
           00000000#  00720                .LONG   0[2]
            00000000  00728                .LONG   0
            00000000  0072C                .LONG   0
           00000000#  00730                .LONG   0[2]
                  1D  00738  INFO_XABFHC:
                                           .BYTE   29
                  2C  00739                .BYTE   44
                0000  0073A                .WORD   0
            00000000  0073C                .LONG   0
           00000000#  00740                .LONG   0[9]
                  02  00764  INFO_NAM:
                                           .BYTE   2
                  60  00765                .BYTE   96
                  00  00766                .BYTE   0
                  00  00767                .BYTE   0
            00000000  00768                .LONG   0
                  00  0076C                .BYTE   0
                  00  0076D                .BYTE   0
                  00  0076E                .BYTE   0
                  00  0076F                .BYTE   0
            00000000  00770                .LONG   0
            00000000  00774                .LONG   0
               0000#  00778                .WORD   0[8]
               0000#  00788                .WORD   0[3]
               0000#  0078E                .WORD   0[3]
            00000000  00794                .LONG   0
            00000000  00798                .LONG   0
                  00  0079C                .BYTE   0
                  00  0079D                .BYTE   0
                  00  0079E                .BYTE   0
                  00  0079F                .BYTE   0
                  00  007A0                .BYTE   0
                  00  007A1                .BYTE   0
```

```
      00# 007A2                .BYTE    0[2]
 00000000 007A4                .LONG    0
 00000000 007A8                .LONG    0
 00000000 007AC                .LONG    0
 00000000 007B0                .LONG    0
 00000000 007B4                .LONG    0
 00000000 007B8                .LONG    0
 00000000# 007BC               .LONG    0[2]
       03 007C4 INFO_FAB:
                               .BYTE    3
       50 007C5                .BYTE    80
     0000 007C6                .WORD    0
 01000000 007C8                .LONG    16777216
 00000000 007CC                .LONG    0
 00000000 007D0                .LONG    0
 00000000 007D4                .LONG    0
     0000 007D8                .WORD    0
       02 007DA                .BYTE    2
       43 007DB                .BYTE    67
 00000000 007DC                .LONG    0
       00 007E0                .BYTE    0
       00 007E1                .BYTE    0
       00 007E2                .BYTE    0
       02 007E3                .BYTE    2
 00000000 007E4                .LONG    0
 00000000 007E8                .LONG    0
 00000000' 007EC               .ADDRESS INFO_NAM
 00000000 007F0                .LONG    0
 00000000 007F4                .LONG    0
       00 007F8                .BYTE    0
       00 007F9                .BYTE    0
     0000 007FA                .WORD    0
 00000000 007FC                .LONG    0
     0000 00800                .WORD    0
       00 00802                .BYTE    0
       00 00803                .BYTE    0
 00000000 00804                .LONG    0
 00000000 00808                .LONG    0
     0000 0080C                .WORD    0
       00 0080E                .BYTE    0
       00 0080F                .BYTE    0
 00000000 00810                .LONG    0
          00814 DISPLAY_WIDTH:
                               .BLKB    4
          00818 FILENAME_WIDTH:
                               .BLKB    4
          0081C OWNER_WIDTH:
                               .BLKB    4
          00820 SIZE_WIDTH:
                               .BLKB    4
          00824 MIN_BLOCK:
                               .BLKB    4
          00828 MAX_BLOCK:
                               .BLKB    4
          0082C ACL_LENGTH:
                               .BLKB    4
          00830 OUTPUT_RAB:
```

DI
V0

```
DIRECTORY                                                   K 15
V04-000                                                     15-Sep-1984 23:38:58    VAX-11 Bliss-32 V4.0-742          Page  20
                                                            14-Sep-1984 12:19:31    [DIR.SRC]DIRECTORY.B32;1                (4)

                                                                   .BLKB    68

                                                                   .PSECT   $PLIT$,NOWRT,NOEXE,2

                        00 00 00 2A 3B 2A 2E 2A  00000 P.AAA:   .ASCII   \*.*;*\<0><0><0>
00 00 53 49 4C 2E 59    52 4F 54 43 45 52 49 44  00008 P.AAB:   .ASCII   \DIRECTORY.LIS\<0><0><0>
                                             00  00017
                                 45 52 4F 46 45 42  00018 P.AAD:   .ASCII   \BEFORE\
                                                 0001E          .BLKB    2
                                        00000006  00020 P.AAC:   .LONG    6
                                        00000000' 00024          .ADDRESS P.AAD
                                    45 43 4E 49 53  00028 P.AAF:   .ASCII   \SINCE\
                                                 0002D          .BLKB    3
                                        00000005  00030 P.AAE:   .LONG    5
                                        00000000' 00034          .ADDRESS P.AAF
                        52 45 4E 57 4F 5F 59 42  00038 P.AAH:   .ASCII   \BY_OWNER\
                                        00000008  00040 P.AAG:   .LONG    8
                                        00000000' 00044          .ADDRESS P.AAH
                                          4C 43 41  00048 P.AAJ:   .ASCII   \ACL\
                                                 0004B          .BLKB    1
                                        00000003  0004C P.AAI:   .LONG    3
                                        00000000' 00050          .ADDRESS P.AAJ
                                    46 45 49 52 42  00054 P.AAL:   .ASCII   \BRIEF\
                                                 00059          .BLKB    3
                                        00000005  0005C P.AAK:   .LONG    5
                                        00000000' 00060          .ADDRESS P.AAL
                              53 4E 4D 55 4C 4F 43  00064 P.AAN:   .ASCII   \COLUMNS\
                                                 0006B          .BLKB    1
                                        00000007  0006C P.AAM:   .LONG    7
                                        00000000' 00070          .ADDRESS P.AAN
                              53 4E 4D 55 4C 4F 43  00074 P.AAP:   .ASCII   \COLUMNS\
                                                 0007B          .BLKB    1
                                        00000007  0007C P.AAO:   .LONG    7
                                        00000000' 00080          .ADDRESS P.AAP
                                       45 54 41 44  00084 P.AAR:   .ASCII   \DATE\
                                        00000004  00088 P.AAQ:   .LONG    4
                                        00000000' 0008C          .ADDRESS P.AAR
                        4C 4C 41 2E 45 54 41 44  00090 P.AAT:   .ASCII   \DATE.ALL\
                                        00000008  00098 P.AAS:   .LONG    8
                                        00000000' 0009C          .ADDRESS P.AAT
            44 45 54 41 45 52 43 2E 45 54 41 44  000A0 P.AAV:   .ASCII   \DATE.CREATED\
                                        0000000C  000AC P.AAU:   .LONG    12
                                        00000000' 000B0          .ADDRESS P.AAV
            44 45 52 49 50 58 45 2E 45 54 41 44  000B4 P.AAX:   .ASCII   \DATE.EXPIRED\
                                        0000000C  000C0 P.AAW:   .LONG    12
                                        00000000' 000C4          .ADDRESS P.AAX
         44 45 49 46 49 44 4F 4D 2E 45 54 41 44  000C8 P.AAZ:   .ASCII   \DATE.MODIFIED\
                                                 000D5          .BLKB    3
                                        0000000D  000D8 P.AAY:   .LONG    13
                                        00000000' 000DC          .ADDRESS P.AAZ
               50 55 4B 43 41 42 2E 45 54 41 44  000E0 P.ABB:   .ASCII   \DATE.BACKUP\
                                                 000EB          .BLKB    1
                                        0000000B  000EC P.ABA:   .LONG    11
                                        00000000' 000F0          .ADDRESS P.ABB
                              44 49 5F 45 4C 49 46  000F4 P.ABD:   .ASCII   \FILE_ID\
                                                 000FB          .BLKB    1
                                        00000007  000FC P.ABC:   .LONG    7
```

```
DIRECTORY                                          L 15
V04-000                                   15-Sep-1984 23:38:58    VAX-11 Bliss-32 V4.0-742          Page  21
                                          14-Sep-1984 12:19:31    [DIR.SRC]DIRECTORY.B32;1                (4)

                                                 00000000'  00100            .ADDRESS P.ABD
                                            4C  4C  55  46  00104  P.ABF:    .ASCII   \FULL\
                                                 00000004.  00108  P.ABE:    .LONG    4
                                                 00000000'  0010C            .ADDRESS P.ABF
                4C  41  54  4F  54  5F  44  4E  41  52  47  00110  P.ABH:    .ASCII   \GRAND_TOTAL\
                                                            0011B            .BLKB    1
                                                 0000000B.  0011C  P.ABG:    .LONG    11
                                                 00000000'  00120            .ADDRESS P.ABH
                                47  4E  49  44  41  45  48  00124  P.ABJ:    .ASCII   \HEADING\
                                                            0012B            .BLKB    1
                                                 00000007.  0012C  P.ABI:    .LONG    7
                                                 00000000'  00130            .ADDRESS P.ABJ
                                52  45  54  4E  49  52  50  00134  P.ABL:    .ASCII   \PRINTER\
                                                            0013B            .BLKB    1
                                                 00000007.  0013C  P.ABK:    .LONG    7
                                                 00000000'  00140            .ADDRESS P.ABL
                                    54  55  50  54  55  4F  00144  P.ABN:    .ASCII   \OUTPUT\
                                                            0014A            .BLKB    2
                                                 00000006.  0014C  P.ABM:    .LONG    6
                                                 00000000'  00150            .ADDRESS P.ABN
                                    54  55  50  54  55  4F  00154  P.ABP:    .ASCII   \OUTPUT\
                                                            0015A            .BLKB    2
                                                 00000006.  0015C  P.ABO:    .LONG    6
                                                 00000000'  00160            .ADDRESS P.ABP
            00  3A  54  55  50  54  55  4F  24  53  59  53  00164  P.ABQ:    .ASCII   \SYS$OUTPUT:\<0>
                                            00  3A  4C  4E  00170  P.ABR:    .ASCII   \NL:\<0>
                                        52  45  4E  57  4F  00174  P.ABT:    .ASCII   \OWNER\
                                                            00179            .BLKB    3
                                                 00000005.  0017C  P.ABS:    .LONG    5
                                                 00000000'  00180            .ADDRESS P.ABT
45  49  46  49  54  4E  45  44  49  2E  52  45  4E  57  4F  00184  P.ABV:    .ASCII   \OWNER.IDENTIFIER\
                                                        52  00193
                                                 00000010.  00194  P.ABU:    .LONG    16
                                                 00000000'  00198            .ADDRESS P.ABV
                    4E  4F  49  54  43  45  54  4F  52  50  0019C  P.ABX:    .ASCII   \PROTECTION\
                                                            001A6            .BLKB    2
                                                 0000000A.  001A8  P.ABW:    .LONG    10
                                                 00000000'  001AC            .ADDRESS P.ABX
                            59  54  49  52  55  43  45  53  001B0  P.ABZ:    .ASCII   \SECURITY\
                                                 00000008.  001B8  P.ABY:    .LONG    8
                                                 00000000'  001BC            .ADDRESS P.ABZ
                                    54  43  45  4C  45  53  001C0  P.ACB:    .ASCII   \SELECT\
                                                            001C6            .BLKB    2
                                                 00000006.  001C8  P.ACA:    .LONG    6
                                                 00000000'  001CC            .ADDRESS P.ACB
4E  49  4D  2E  45  5A  49  53  2E  54  43  45  4C  45  53  001D0  P.ACD:    .ASCII   \SELECT.SIZE.MINIMUM_SIZE\
                        45  5A  49  53  5F  4D  55  4D  49  001DF
                                                 00000018.  001E8  P.ACC:    .LONG    24
                                                 00000000'  001EC            .ADDRESS P.ACD
4E  49  4D  2E  45  5A  49  53  2E  54  43  45  4C  45  53  001F0  P.ACF:    .ASCII   \SELECT.SIZE.MINIMUM_SIZE\
                        45  5A  49  53  5F  4D  55  4D  49  001FF
                                                 00000018.  00208  P.ACE:    .LONG    24
                                                 00000000'  0020C            .ADDRESS P.ACF
58  41  4D  2E  45  5A  49  53  2E  54  43  45  4C  45  53  00210  P.ACH:    .ASCII   \SELECT.SIZE.MAXIMUM_SIZE\
                        45  5A  49  53  5F  4D  55  4D  49  0021F
                                                 00000018.  00228  P.ACG:    .LONG    24
                                                 00000000'  0022C            .ADDRESS P.ACH
```

```
5B 41 4D 2E 45 5A 49 53 2E 54 43 45 4C 45 53  00230 P.ACJ:  .ASCII  \SELECT.SIZE.MAXIMUM_SIZE\
                  45 5A 49 53 5F 4D 55 4D 49  0023F
                                    00000018  00248 P.ACI:  .LONG   24
                                    00000000' 0024C         .ADDRESS P.ACJ
                                 45 5A 49 53  00250 P.ACL:  .ASCII  \SIZE\
                                    00000004  00254 P.ACK:  .LONG   4
                                    00000000' 00258         .ADDRESS P.ACL
                     4C 4C 41 2E 45 5A 49 53  0025C P.ACN:  .ASCII  \SIZE.ALL\
                                    00000008  00264 P.ACM:  .LONG   8
                                    00000000' 00268         .ADDRESS P.ACN
4E 4F 49 54 41 43 4F 4C 4C 41 2E 45 5A 49 53  0026C P.ACP:  .ASCII  \SIZE.ALLOCATION\
                                              0027B         .BLKB   1
                                    0000000F  0027C P.ACO:  .LONG   15
                                    00000000' 00280         .ADDRESS P.ACP
                  44 45 53 55 2E 45 5A 49 53  00284 P.ACR:  .ASCII  \SIZE.USED\
                                              0028D         .BLKB   3
                                    00000009  00290 P.ACQ:  .LONG   9
                                    00000000' 00294         .ADDRESS P.ACR
                              4C 41 54 4F 54  00298 P.ACT:  .ASCII  \TOTAL\
                                              0029D         .BLKB   3
                                    00000005  002A0 P.ACS:  .LONG   5
                                    00000000' 002A4         .ADDRESS P.ACT
                     47 4E 49 4C 49 41 52 54  002A8 P.ACV:  .ASCII  \TRAILING\
                                    00000008  002B0 P.ACU:  .LONG   8
                                    00000000' 002B4         .ADDRESS P.ACV
                     53 4E 4F 49 53 52 45 56  002B8 P.ACX:  .ASCII  \VERSIONS\
                                    00000008  002C0 P.ACW:  .LONG   8
                                    00000000' 002C4         .ADDRESS P.ACX
                     53 4E 4F 49 53 52 45 56  002C8 P.ACZ:  .ASCII  \VERSIONS\
                                    00000008  002D0 P.ACY:  .LONG   8
                                    00000000' 002D4         .ADDRESS P.ACZ
                              48 54 44 49 57  002D8 P.ADB:  .ASCII  \WIDTH\
                                              002DD         .BLKB   3
                                    00000005  002E0 P.ADA:  .LONG   5
                                    00000000' 002E4         .ADDRESS P.ADB
      59 41 4C 50 53 49 44 2E 48 54 44 49 57  002E8 P.ADD:  .ASCII  \WIDTH.DISPLAY\
                                              002F5         .BLKB   3
                                    0000000D  002F8 P.ADC:  .LONG   13
                                    00000000' 002FC         .ADDRESS P.ADD
   45 4D 41 4E 45 4C 49 46 2E 48 54 44 49 57  00300 P.ADF:  .ASCII  \WIDTH.FILENAME\
                                              0030E         .BLKB   2
                                    0000000E  00310 P.ADE:  .LONG   14
                                    00000000' 00314         .ADDRESS P.ADF
         52 45 4E 57 4F 2E 48 54 44 49 57  00318 P.ADH:  .ASCII  \WIDTH.OWNER\
                                              00323         .BLKB   1
                                    0000000B  00324 P.ADG:  .LONG   11
                                    00000000' 00328         .ADDRESS P.ADH
            45 5A 49 53 2E 48 54 44 49 57  0032C P.ADJ:  .ASCII  \WIDTH.SIZE\
                                              00336         .BLKB   2
                                    0000000A  00338 P.ADI:  .LONG   10
                                    00000000' 0033C         .ADDRESS P.ADJ
            54 55 50 54 55 4F 24 53 59 53  00340 P.ADL:  .ASCII  \SYS$OUTPUT\
                                              0034A         .BLKB   2
                                    0000000A  0034C P.ADK:  .LONG   10
                                    00000000' 00350         .ADDRESS P.ADL
                              54 55 50 4E 49  00354 P.ADN:  .ASCII  \INPUT\
                                              00359         .BLKB   3
```

```
                                                                   N 15
DIRECTORY                                                          15-Sep-1984 23:38:58    VAX-11 Bliss-32 V4.0-742              Page  23
V04-000                                                            14-Sep-1984 12:19:31    [DIR.SRC]DIRECTORY.B32;1                    (4)

                                          00000005  0035C P.ADM:   .LONG    5                                                   ;
                                          00000000' 00360          .ADDRESS P.ADN                                               ;
                    52  48  53  45  52  55  43  45  53  00364 P.ADP:   .ASCII   \SECURESHR\                                         ;
                                                    0036D          .BLKB    3                                                   ;
                                          00000009  00370 P.ADO:   .LONG    9                                                   ;
                                          00000000' 00374          .ADDRESS P.ADP                                               ;
    4C  43  41  5F  54  41  4D  52  4F  46  24  53  59  53  00378 P.ADR:   .ASCII   \SYS$FORMAT_ACL\                                    ;
                                                    00386          .BLKB    2                                                   ;
                                          0000000E  00388 P.ADQ:   .LONG    14                                                  ;
                                          00000000' 0038C          .ADDRESS P.ADR                                               ;

                                                                   .PSECT   $OWN$,NOEXE,2

                                                    00000 FORMAT_ACL_ADDR:
                                                                   .BLKB    4
                                                    00004 OUTPUT_FAB:
                                                                   .BLKB    80
                                                    00054 OUTPUT_NAM:
                                                                   .BLKB    96
                                                    000B4 OUT_EXP_NAM:
                                                                   .BLKB    255
                                                    001B3          .BLKB    1
                                                    001B4 OUT_RES_NAM:
                                                                   .BLKB    255

                                                          $RMS_PTR=         OUTPUT_FAB
                                                          $RMS_PTR=         OUTPUT_RAB
                                                          $RMS_PTR=         OUTPUT_NAM
                                                                   .EXTRN   CLI$GET_VALUE, CLI$PRESENT
                                                                   .EXTRN   LIB$FILE_SCAN, LIB$FIND_IMAGE_SYMBOL
                                                                   .EXTRN   LIB$QUAL_FILE_PARSE
                                                                   .EXTRN   CLI$_DEFAULTED, CLI$_NEGATED
                                                                   .EXTRN   DIR$GET_INFO, DIR$TOTAL
                                                                   .EXTRN   DIR$GRAND_TOTAL
                                                                   .EXTRN   LIB$CVT_DTB, LIB$GET_VM
                                                                   .EXTRN   DIR$_NOFILES, LIB$SIGNAL
                                                                   .EXTRN   SYS$FLUSH, SYS$WAIT
                                                                   .EXTRN   SYS$CREATE, SYS$CONNECT
                                                                   .EXTRN   SYS$GETDVI, SYS$CLOSE

                                                                   .PSECT   $CODE$,NOWRT,2

                                               0FFC 00000 DIR$MAIN:
                                                                   .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 0591
                                 5B      0000'   CF   9E 00002     MOVAB    $RMS_PTR, R11                                   ;
                                 5A      0000'   CF   9E 00007     MOVAB    P.AAX, R10                                      ;
                                 59 00000000G    00   9E 0000C     MOVAB    CLI$PRESENT, R9                                 ;
                                 58 00000000'    EF   9E 00013     MOVAB    QUAL_FLAGS, R8                                  ;
                                 5E      FD14    CE   9E 0001A     MOVAB    -748(SP), SP                                    ;
                                         0C      AE   D4 0001F     CLRL     SCAN_CONTEXT                                    ; 0660
                                                 68   D4 00022     CLRL     QUAL_FLAGS                                      ; 0661
                          14     A8              01   D0 00024     MOVL     #1, WORST_ERROR                                 ; 0662
                                         20      A8   D4 00028     CLRL     CHANNEL                                         ; 0663
          10              00     6E              00   2C 0002B     MOVC5    #0, (SP), #0, #16, DEVICE_NAME                  ; 0664
                                         24      A8      00030                                                              ;
                                         0C      A8   7C 00032     CLRQ     COLUMN_INDEX                                    ; 0665
                                         08      A8   D4 00035     CLRL     COLUMN_COUNT                                    ;
```

```
                                           0660  C8  7C 00038          CLRQ    VERSION_COUNT                              ; 0666
                                           065C  C8  D4 0003C          CLRL    PREV_FILE_LEN                              ; 0667
                                           0558  C8  D4 00040          CLRL    PREV_DIR_LEN                               ;
                                           0440  C8  7C 00044          CLRQ    TOTAL_ALLOC                                ; 0668
                                           043C  C8  D4 00048          CLRL    TOTAL_USED                                 ;
                                           0450  C8  7C 0004C          CLRQ    GRAND_FILES                                ; 0669
                                           0448  C8  7C 00050          CLRQ    GRAND_USED                                 ;
                                             10  A8  D4 00054          CLRL    COLUMN_WIDTH                               ; 0670
                                             04  AE  7C 00057          CLRQ    INDEV_CLASS                                ; 0671
                                                 56  D4 0005A          CLRL    XAB_PTR                                    ; 0672
                                           066B  C8  D4 0005C          CLRL    FIRST_XAB                                  ;
        08              00            6E         00  2C 00060          MOVC5   #0, (SP), #0, #8, VALUE_DESC               ; 0673
                                             2C  AE     00065
                              2F      AE         02  90 00067          MOVB    #2, VALUE_DESC+3                           ; 0674
              34   AE         2C      AE         08  28 0006B          MOVC3   #8, VALUE_DESC, FILE_DESC                  ; 0675
              34   A8         2C      AE         08  28 00071          MOVC3   #8, VALUE_DESC, LINE_DESC                  ; 0676
                              38      A8     3C  A8  9E 00077          MOVAB   LINE_BUFFER, LINE_DESC+4                   ; 0677
                                             1C  A8  9F 0007C          PUSHAB  DISPLAY_BLOCK                              ; 0681
                              04      AE   01CB  8F  3C 0007F          MOVZWL  #459, 4(SP)                                ;
                                             04  AE  9F 00085          PUSHAB  4(SP)                                      ;
                       00000000G  00             02  FB 00088          CALLS   #2, LIB$GET_VM                             ;
                                  57             50  D0 0008F          MOVL    R0, STATUS                                 ;
                                  3D             57  E8 00092          BLBS    STATUS, 4$                                 ; 0682
                                           0830  C8  9F 00095 1$:      PUSHAB  OUTPUT_RAB                                 ; 0685
                       00000000G  00             01  FB 00099          CALLS   #1, SYS$FLUSH                              ;
                                           0830  C8  9F 000A0          PUSHAB  OUTPUT_RAB                                 ;
                       00000000G  00             01  FB 000A4          CALLS   #1, SYS$WAIT                               ;
                                                 57  DD 000AB          PUSHL   STATUS                                     ;
                       00000000G  00             01  FB 000AD          CALLS   #1, LIB$SIGNAL                             ;
                                  07             57  93 000B4          BITB    STATUS, #7                                 ;
                                                 16  13 000B7          BEQL    3$                                         ;
        50              57        03             00  EF 000B9          EXTZV   #0, #3, STATUS, R0                         ;
        50        14    A8        03             00  ED 000BE          CMPZV   #0, #3, WORST_ERROR, R0                    ;
                                                 09  18 000C4          BGEQ    3$                                         ;
                  14    A8        57 10000000    8F  C9 000C6 2$:      BISL3   #268435456, STATUS, WORST_ERROR            ;
                                               087F  31 000CF 3$:      BRW     86$                                        ; 0686
   0050 8F              00            6E         00  2C 000D2 4$:      MOVC5   #0, (SP), #0, #80, $RMS_PTR                ; 0694
                                             B0  AD     000D9
                              B0      AD   5003  8F  B0 000DB          MOVW    #20483, $RMS_PTR                           ;
                              C6      AD         02  90 000E1          MOVB    #2, $RMS_PTR+22                            ;
                              CF      AD         02  90 000E5          MOVB    #2, $RMS_PTR+31                            ;
                              D8      AD   FF50  CD  9E 000E9          MOVAB   INPUT_NAM, $RMS_PTR+40                     ;
                              E0      AD         6A  9E 000EF          MOVAB   P_AAA, $RMS_PTR+48                         ;
                              E5      AD         05  90 000F3          MOVB    #5, $RMS_PTR+53                            ;
   0060 8F              00            6E         00  2C 000F7          MOVC5   #0, (SP), #0, #96, $RMS_PTR                ; 0699
                                           FF50  CD     000FE
                            FF50      CD   6002  8F  B0 00101          MOVW    #24578, $RMS_PTR                           ;
                            FF52      CD         01  8E 00108          MNEGB   #1, $RMS_PTR+2                             ;
                            FF54      CD     3C  AE  9E 0010D          MOVAB   INP_RES_NAM, $RMS_PTR+4                    ;
                            FF5A      CD         01  8E 00113          MNEGB   #1, $RMS_PTR+10                            ;
                            FF5C      CD   013C  CE  9E 00118          MOVAB   INP_EXP_NAM, $RMS_PTR+12                   ;
   0050 8F              00            6E         00  2C 0011F          MOVC5   #0, (SP), #0, #80, $RMS_PTR                ; 0707
                                                 6B     00126
                                      6B   5003  8F  B0 00127          MOVW    #20483, $RMS_PTR                           ;
                              04      AB     40  8F  9A 0012C          MOVZBL  #64, $RMS_PTR+4                            ;
                              16      AB         01  90 00131          MOVB    #1, $RMS_PTR+22                            ;
                              1E      AB   0202  8F  B0 00135          MOVW    #514, $RMS_PTR+30                          ;
                              28   AB        50   AB   9E 0013B        MOVAB   OUTPUT_NAM, $RMS_PTR+40              :
                              30   AB        08   AA   9E 00140        MOVAB   P.AAB, $RMS_PTR+48                   :
                              35   AB             0D   90 00145        MOVB    #13, $RMS_PTR+53                     :
      0044   8F        00          6E             00   2C 00149        MOVC5   #0, (SP), #0, #68, $RMS_PTR          : 0709
                                           0830   C8      00150                                                     :
                            0830   C8      4401   8F   B0 00153        MOVW    #17409, $RMS_PTR                     :
                            086C   C8             6B   9E 0015A        MOVAB   OUTPUT_FAB, $RMS_PTR+60              :
      0060   8F        00          6E             00   2C 0015F        MOVC5   #0, (SP), #0, #96, $RMS_PTR          : 0714
                                             50   AB      00166                                                     :
                              50   AB      6002   8F   B0 00168        MOVW    #24578, $RMS_PTR                     :
                              52   AB             01   8E 0016E        MNEGB   #1, $RMS_PTR+2                       :
                              54   AB      01B0   CB   9E 00172        MOVAB   OUT_RES_NAM, $RMS_PTR+4              :
                              5A   AB             01   8E 00178        MNEGB   #1, $RMS_PTR+10                      :
                              5C   AB      00B0   CB   9E 0017C        MOVAB   OUT_EXP_NAM, $RMS_PTR+12             :
                                             20   AA   9F 00182        PUSHAB  P.AAC                                : 0722
                                   69             01   FB 00185        CALLS   #1, CLI$PRESENT                      :
                                   09             50   E8 00188        BLBS    R0, 5$                               :
                                             30   AA   9F 0018B        PUSHAB  P.AAE                                : 0723
                                   69             01   FB 0018E        CALLS   #1, CLI$PRESENT                      :
                                   06             50   E9 00191        BLBC    R0, 6$                               :
                              03   A8      0240   8F   A8 00194 5$:    BISW2   #576, QUAL_FLAGS+3                   : 0726
                                             40   AA   9F 0019A 6$:    PUSHAB  P.AAG                                : 0730
                                   69             01   FB 0019D        CALLS   #1, CLI$PRESENT                      :
                                   06             50   E9 001A0        BLBC    R0, 7$                               :
                              03   A8      0440   8F   A8 001A3        BISW2   #1088, QUAL_FLAGS+3                  : 0733
                                             4C   AA   9F 001A9 7$:    PUSHAB  P.AAI                                : 0739
                                   69             01   FB 001AC        CALLS   #1, CLI$PRESENT                      :
             68        01          00             50   F0 001AF        INSV    R0, #0, #1, QUAL_FLAGS               :
                                             5C   AA   9F 001B4        PUSHAB  P.AAK                                : 0740
                                   69             01   FB 001B7        CALLS   #1, CLI$PRESENT                      :
             68        01          01             50   F0 001BA        INSV    R0, #1, #1, QUAL_FLAGS               :
                                             6C   AA   9F 001BF        PUSHAB  P.AAM                                : 0741
                                   69             01   FB 001C2        CALLS   #1, CLI$PRESENT                      :
             68        01          02             50   F0 001C5        INSV    R0, #2, #1, QUAL_FLAGS               :
                                   52             50   D0 001CA        MOVL    R0, CLI_STATUS                       :
                                   40             50   E9 001CD        BLBC    R0, 11$                              :
                                             2C   AE   9F 001D0        PUSHAB  VALUE_DESC                           : 0744
                                             7C   AA   9F 001D3        PUSHAB  P.AAO                                :
                       00000000G   00             02   FB 001D6        CALLS   #2, CLI$GET_VALUE                    :
                                             08   A8   9F 001DD        PUSHAB  COLUMN_COUNT                         : 0745
                                             34   AE   DD 001E0        PUSHL   VALUE_DESC+4                         : 0746
                                   7E        34   AE   3C 001E3        MOVZWL  VALUE_DESC, -(SP)                    : 0745
                       00000000G   00             03   FB 001E7        CALLS   #3, LIB$CVT_DTB                      :
                                   57             50   D0 001EE        MOVL    R0, STATUS                           :
                                   03             57   E8 001F1        BLBS    STATUS, 9$                           : 0748
                                                0393   31 001F4 8$:    BRW     40$                                  :
                                             08   A8   D5 001F7 9$:    TSTL    COLUMN_COUNT                         :
                                                  F8   19 001FA        BLSS    8$                                   :
                                                  04   12 001FC        BNEQ    10$                                  : 0754
                              08   A8             01   D0 001FE        MOVL    #1, COLUMN_COUNT                     :
                       00000000G   8F             52   D1 00202 10$:   CMPL    CLI_STATUS, #CLI$_DEFAULTED          : 0755
                                                  05   12 00209        BNEQ    11$                                  :
                              03   A8        80   8F   88 0020B        BISB2   #128, QUAL_FLAGS+3                   :
                                           0088   CA   9F 00210 11$:   PUSHAB  P.AAQ                                : 0757
                                   69             01   FB 00214        CALLS   #1, CLI$PRESENT                      :
             68        01          03             50   F0 00217        INSV    R0, #3, #1, QUAL_FLAGS               :
                                   62             50   E9 0021C        BLBC    R0, 16$                              :
```

```
                          04  A8            02  88 0021F        BISB2   #2, QUAL_FLAGS+4                        : 0760
                                   0098     CA  9F 00223        PUSHAB  P.AAS                                   : 0761
                              69            01  FB 00227        CALLS   #1, CLI$PRESENT
                              0E            50  E9 0022A        BLBC    R0, 12$
                              68       F0   8F  88 0022D        BISB2   #240, QUAL_FLAGS                        : 0767
                          10  A8 0000004C   8F  C0 00231        ADDL2   #76, COLUMN_WIDTH                       : 0768
                                            46  11 00239        BRB     16$                                     : 0761
                                   00AC     CA  9F 0023B 12$:   PUSHAB  P.AAU                                   : 0772
                              69            01  FB 0023F        CALLS   #1, CLI$PRESENT
                              07            50  E9 00242        BLBC    R0, 13$
                              68            10  88 00245        BISB2   #16, QUAL_FLAGS                         : 0775
                          10  A8            13  C0 00248        ADDL2   #19, COLUMN_WIDTH                       : 0776
                                   00C0     CA  9F 0024C 13$:   PUSHAB  P.AAW                                   : 0778
                              69            01  FB 00250        CALLS   #1, CLI$PRESENT
                              07            50  E9 00253        BLBC    R0, 14$
                              68            20  88 00256        BISB2   #32, QUAL_FLAGS                         : 0781
                          10  A8            13  C0 00259        ADDL2   #19, COLUMN_WIDTH                       : 0782
                                   00D8     CA  9F 0025D 14$:   PUSHAB  P.AAY                                   : 0784
                              69            01  FB 00261        CALLS   #1, CLI$PRESENT
                              08            50  E9 00264        BLBC    R0, 15$
                              68       40   8F  88 00267        BISB2   #64, QUAL_FLAGS                         : 0787
                          10  A8            13  C0 0026B        ADDL2   #19, COLUMN_WIDTH                       : 0788
                                   00EC     CA  9F 0026F 15$:   PUSHAB  P.ABA                                   : 0790
                              69            01  FB 00273        CALLS   #1, CLI$PRESENT
                              08            50  E9 00276        BLBC    R0, 16$
                              68       80   8F  88 00279        BISB2   #128, QUAL_FLAGS                        : 0793
                          10  A8            13  C0 0027D        ADDL2   #19, COLUMN_WIDTH                       : 0794
                                   00FC     CA  9F 00281 16$:   PUSHAB  P.ABC                                   : 0798
                              69            01  FB 00285        CALLS   #1, CLI$PRESENT
   01   A8          01        00            50  F0 00288        INSV    R0, #0, #1, QUAL_FLAGS+1
                              04            50  E9 0028E        BLBC    R0, 17$
                          10  A8            15  C0 00291        ADDL2   #21, COLUMN_WIDTH                       : 0799
                                   0108     CA  9F 00295 17$:   PUSHAB  P.ABE                                   : 0800
                              69            01  FB 00299        CALLS   #1, CLI$PRESENT
   01   A8          01        01            50  F0 0029C        INSV    R0, #1, #1, QUAL_FLAGS+1
                              04            50  E9 002A2        BLBC    R0, 18$
                          04  A8            1F  88 002A5        BISB2   #31, QUAL_FLAGS+4                       : 0805
                                   011C     CA  9F 002A9 18$:   PUSHAB  P.ABG                                   : 0807
                              69            01  FB 002AD        CALLS   #1, CLI$PRESENT
   01   A8          01        02            50  F0 002B0        INSV    R0, #2, #1, QUAL_FLAGS+1
                                   012C     CA  9F 002B6        PUSHAB  P.ABI                                   : 0808
                              69            01  FB 002BA        CALLS   #1, CLI$PRESENT
   01   A8          01        03            50  F0 002BD        INSV    R0, #3, #1, QUAL_FLAGS+1
                                   013C     CA  9F 002C3        PUSHAB  P.ABK                                   : 0813
                              69            01  FB 002C7        CALLS   #1, CLI$PRESENT
   01   A8          01        06            50  F0 002CA        INSV    R0, #6, #1, QUAL_FLAGS+1
                              05            50  E9 002D0        BLBC    R0, 19$
                          05  A8       A0   8F  88 002D3        BISB2   #160, OUTPUT_FAB+5                      : 0817
                                   014C     CA  9F 002D8 19$:   PUSHAB  P.ABM                                   : 0819
                              69            01  FB 002DC        CALLS   #1, CLI$PRESENT
   01   A8          01        04            50  F0 002DF        INSV    R0, #4, #1, QUAL_FLAGS+1
                              52            50  D0 002E5        MOVL    R0, CLI_STATUS
                              30            50  E9 002E8        BLBC    R0, 20$
                                     34     AE  9F 002EB        PUSHAB  FILE_DESC                               : 0822
                                   015C     CA  9F 002EE        PUSHAB  P.ABO
                   00000000G  00            02  FB 002F2        CALLS   #2, CLI$GET_VALUE
                          2C  AB     38     AE  D0 002F9        MOVL    FILE_DESC+4, OUTPUT_FAB+44              : 0823
```

```
                                                                                 E 16
DIRECTORY                                                                        15-Sep-1984 23:38:58    VAX-11 Bliss-32 V4.0-742               Page  27
V04-000                                                                          14-Sep-1984 12:19:31    [DIR.SRC]DIRECTORY.B32;1                     (4)

                                  50        34   AE  3C 002FE         MOVZWL  FILE_DESC, R0                                             ; 0824
                             34   AB             50  90 00302         MOVB    R0, OUTPUT_FAB+52                                         ;
                                                 50  D5 00306         TSTL    R0                                                        ;
                                                 29  12 00308         BNEQ    21$                                                       ;
                     24      01   AB             06  E0 0030A         BBS     #6, QUAL_FLAGS+1, 21$                                     ; 0825
                             2C   AB      0164   CA  9E 0030F         MOVAB   P.ABQ, OUTPUT_FAB+44                                      ; 0828
                             34   AB             0B  90 00315         MOVB    #11, OUTPUT_FAB+52                                        ; 0829
                                                 18  11 00319         BRB     21$                                                       ; 0819
                      00000000G   8F             52  D1 0031B  20$:   CMPL    CLI_STATUS, #CLI$_NEGATED                                 ; 0834
                                                 0F  12 00322         BNEQ    21$                                                       ;
                             2C   AB      0170   CA  9E 00324         MOVAB   P.ABR, OUTPUT_FAB+44                                      ; 0837
                             34   AB             03  90 0032A         MOVB    #3, OUTPUT_FAB+52                                         ; 0838
                             05   AB        A0   8F  8A 0032E         BICB2   #160, OUTPUT_FAB+5                                        ; 0840
                                          017C   CA  9F 00333  21$:   PUSHAB  P.ABS                                                     ; 0843
                                  69             01  FB 00337         CALLS   #1, CLI$PRESENT                                           ;
       01   A8       01           05             50  F0 0033A         INSV    R0, #5, #1, QUAL_FLAGS+1                                  ;
                                  11             50  E9 00340         BLBC    R0, 22$                                                   ;
                             04   A8             04  88 00343         BISB2   #4, QUAL_FLAGS+4                                          ; 0846
                                          0194   CA  9F 00347         PUSHAB  P.ABU                                                     ; 0847
                                  69             01  FB 0034B         CALLS   #1, CLI$PRESENT                                           ;
       04   A8       01           06             50  F0 0034E         INSV    R0, #6, #1, QUAL_FLAGS+4                                  ;
                                          01A8   CA  9F 00354  22$:   PUSHAB  P.ABW                                                     ; 0849
                                  69             01  FB 00358         CALLS   #1, CLI$PRESENT                                           ;
       01   A8       01           07             50  F0 0035B         INSV    R0, #7, #1, QUAL_FLAGS+1                                  ;
                                  08             50  E9 00361         BLBC    R0, 23$                                                   ;
                             04   A8             04  88 00364         BISB2   #4, QUAL_FLAGS+4                                          ; 0852
                             10   A8             17  C0 00368         ADDL2   #23, COLUMN_WIDTH                                         ; 0853
                                          01B8   CA  9F 0036C  23$:   PUSHAB  P.ABY                                                     ; 0855
                                  69             01  FB 00370         CALLS   #1, CLI$PRESENT                                           ;
       02   A8       01           00             50  F0 00373         INSV    R0, #0, #1, QUAL_FLAGS+2                                  ;
                                  0F             50  E9 00379         BLBC    R0, 24$                                                   ;
                             01   A8  040000A0   8F  C8 0037C         BISL2   #67109024, QUAL_FLAGS+1                                   ; 0860
                                  68             01  88 00384         BISB2   #1, QUAL_FLAGS                                            ; 0859
                             10   A8             17  C0 00387         ADDL2   #23, COLUMN_WIDTH                                         ; 0861
                                          01C8   CA  9F 0038B  24$:   PUSHAB  P.ACA                                                     ; 0863
                                  69             01  FB 0038F         CALLS   #1, CLI$PRESENT                                           ;
                                  52             50  E9 00392         BLBC    R0, 26$                                                   ;
                                          0824   C8  D4 00395         CLRL    MIN_BLOCK                                                 ; 0866
                           0828   C8  3FFFFFFF   8F  D0 00399         MOVL    #1073741823, MAX_BLOCK                                    ; 0867
                                          01E8   CA  9F 003A2         PUSHAB  P.ACC                                                     ; 0868
                                  69             01  FB 003A6         CALLS   #1, CLI$PRESENT                                           ;
                                  34             50  E9 003A9         BLBC    R0, 25$                                                   ;
                             02   A8             04  88 003AC         BISB2   #4, QUAL_FLAGS+2                                          ; 0871
                                            2C   AE  9F 003B0         PUSHAB  VALUE_DESC                                                ; 0872
                                          0208   CA  9F 003B3         PUSHAB  P.ACE                                                     ;
                      00000000G   00             02  FB 003B7         CALLS   #2, CLI$GET_VALUE                                         ;
                                          0824   C8  9F 003BE         PUSHAB  MIN_BLOCK                                                 ; 0873
                                            34   AE  DD 003C2         PUSHL   VALUE_DESC+4                                              ; 0874
                                  7E        34   AE  3C 003C5         MOVZWL  VALUE_DESC, -(SP)                                         ; 0873
                      00000000G   00             03  FB 003C9         CALLS   #3, LIB$CVT_DTB                                           ;
                                  57             50  D0 003D0         MOVL    R0, STATUS                                                ;
                                  3B             57  E9 003D3         BLBC    STATUS, 27$                                               ; 0876
                                          0824   C8  D5 003D6         TSTL    MIN_BLOCK                                                 ;
                                                 3F  19 003DA         BLSS    30$                                                       ;
                             04   A8             01  88 003DC         BISB2   #1, QUAL_FLAGS+4                                          ; 0882
                                          0228   CA  9F 003E0  25$:   PUSHAB  P.ACG                                                     ; 0884
                                  69             01  FB 003E4         CALLS   #1, CLI$PRESENT                                           ;
```

```
                                   37               50  E9  003E7  26$:  BLBC    R0, 31$                          ;
                          02       A8               04  88  003EA        BISB2   #4, QUAL_FLAGS+2                 ; 0887
                                             2C     AE  9F  003EE        PUSHAB  VALUE_DESC                       ; 0888
                                           0248     CA  9F  003F1        PUSHAB  P.ACI                            ;
                   00000000G       00               02  FB  003F5        CALLS   #2, CLI$GET_VALUE                ;
                                           0828     C8  9F  003FC        PUSHAB  MAX_BLOCK                        ; 0889
                                             34     AE  DD  00400        PUSHL   VALUE_DESC+4                     ; 0890
                                   7E        34     AE  3C  00403        MOVZWL  VALUE_DESC, -(SP)                ; 0889
                   00000000G       00               03  FB  00407        CALLS   #3, LIB$CVT_DTB                  ;
                                   57               50  D0  0040E        MOVL    R0, STATUS                       ;
                                   03               57  E8  00411  27$:  BLBS    STATUS, 29$                      ; 0892
                                                  0173  31  00414  28$:  BRW     40$                              ;
                                           0828     C8  D5  00417  29$:  TSTL    MAX_BLOCK                        ;
                                                    F7  19  0041B  30$:  BLSS    28$                              ;
                          04       A8               01  88  0041D        BISB2   #1, QUAL_FLAGS+4                 ; 0898
                                           0254     CA  9F  00421  31$:  PUSHAB  P.ACK                            ; 0901
                                   69               01  FB  00425        CALLS   #1, CLI$PRESENT                  ;
      02    A8           01        03               50  F0  00428        INSV    R0, #3, #1, QUAL_FLAGS+2         ;
                                   2F               50  E9  0042E        BLBC    R0, 34$                          ;
                          04       A8               01  88  00431        BISB2   #1, QUAL_FLAGS+4                 ; 0904
                                           0264     CA  9F  00435        PUSHAB  P.ACM                            ; 0905
                                   69               01  FB  00439        CALLS   #1, CLI$PRESENT                  ;
                                   04               50  E9  0043C        BLBC    R0, 32$                          ;
                          02       A8               10  88  0043F        BISB2   #16, QUAL_FLAGS+2                ; 0906
                                           027C     CA  9F  00443  32$:  PUSHAB  P.ACO                            ; 0907
                                   69               01  FB  00447        CALLS   #1, CLI$PRESENT                  ;
                                   04               50  E9  0044A        BLBC    R0, 33$                          ;
                          02       A8               20  88  0044D        BISB2   #32, QUAL_FLAGS+2                ; 0908
                                           0290     CA  9F  00451  33$:  PUSHAB  P.ACQ                            ; 0909
                                   69               01  FB  00455        CALLS   #1, CLI$PRESENT                  ;
                                   05               50  E9  00458        BLBC    R0, 34$                          ;
                          02       A8        40     8F  88  0045B        BISB2   #64, QUAL_FLAGS+2                ; 0910
                                           02A0     CA  9F  00460  34$:  PUSHAB  P.ACS                            ; 0912
                                   69               01  FB  00464        CALLS   #1, CLI$PRESENT                  ;
      02    A8           01        07               50  F0  00467        INSV    R0, #7, #1, QUAL_FLAGS+2         ;
                                           02B0     CA  9F  0046D        PUSHAB  P.ACU                            ; 0913
                                   69               01  FB  00471        CALLS   #1, CLI$PRESENT                  ;
      03    A8           01        00               50  F0  00474        INSV    R0, #0, #1, QUAL_FLAGS+3         ;
                                           02C0     CA  9F  0047A        PUSHAB  P.ACW                            ; 0914
                                   69               01  FB  0047E        CALLS   #1, CLI$PRESENT                  ;
      03    A8           01        01               50  F0  00481        INSV    R0, #1, #1, QUAL_FLAGS+3         ;
                                   2F               50  E9  00487        BLBC    R0, 36$                          ;
                                             2C     AE  9F  0048A        PUSHAB  VALUE_DESC                       ; 0917
                                           02D0     CA  9F  0048D        PUSHAB  P.ACY                            ;
                   00000000G       00               02  FB  00491        CALLS   #2, CLI$GET_VALUE                ;
                                           0660     C8  9F  00498        PUSHAB  VERSION_COUNT                    ; 0918
                                             34     AE  DD  0049C        PUSHL   VALUE_DESC+4                     ; 0919
                                   7E        34     AE  3C  0049F        MOVZWL  VALUE_DESC, -(SP)                ; 0918
                   00000000G       00               03  FB  004A3        CALLS   #3, LIB$CVT_DTB                  ;
                                   57               50  D0  004AA        MOVL    R0, STATUS                       ;
                                   06               57  E9  004AD        BLBC    STATUS, 35$                      ; 0921
                                           0660     C8  D5  004B0        TSTL    VERSION_COUNT                    ;
                                                    03  14  004B4        BGTR    36$                              ;
                                                  00D1  31  004B6  35$:  BRW     40$                              ;
                                           02E0     CA  9F  004B9  36$:  PUSHAB  P.ADA                            ; 0928
                                   69               01  FB  004BD        CALLS   #1, CLI$PRESENT                  ;
      03    A8           01        02               50  F0  004C0        INSV    R0, #2, #1, QUAL_FLAGS+3         ;
```

```
                  03            50  E8 004C6          BLBS    R0, 37$                          :
                              0100  31 004C9          BRW     43$                              :
                          2C    AE  9F 004CC 37$:     PUSHAB  VALUE_DESC                       : 0931
                        02F8    CA  9F 004CF          PUSHAB  P.ADC                            :
         00000000G  00          02  FB 004D3          CALLS   #2, CLI$GET_VALUE                :
                        0814    C8  9F 004DA          PUSHAB  DISPLAY_WIDTH                    : 0932
                          34    AE  DD 004DE          PUSHL   VALUE_DESC+4                     : 0933
                  7E      34    AE  3C 004E1          MOVZWL  VALUE_DESC, -(SP)                : 0932
         00000000G  00          03  FB 004E5          CALLS   #3, LIB$CVT_DTB                  :
                  57            50  D0 004EC          MOVL    R0, STATUS                       :
                  C4            57  E9 004EF          BLBC    STATUS, 35$                      : 0935
                        0814    C8  D5 004F2          TSTL    DISPLAY_WIDTH                    :
                                BE  19 004F6          BLSS    35$                              :
                          2C    AE  9F 004F8          PUSHAB  VALUE_DESC                       : 0941
                        0310    CA  9F 004FB          PUSHAB  P.ADE                            :
         00000000G  00          02  FB 004FF          CALLS   #2, CLI$GET_VALUE                :
                        0818    C8  9F 00506          PUSHAB  FILENAME_WIDTH                   : 0942
                          34    AE  DD 0050A          PUSHL   VALUE_DESC+4                     : 0943
                  7E      34    AE  3C 0050D          MOVZWL  VALUE_DESC, -(SP)                : 0942
         00000000G  00          03  FB 00511          CALLS   #3, LIB$CVT_DTB                  :
                  57            50  D0 00518          MOVL    R0, STATUS                       :
                  6C            57  E9 0051B          BLBC    STATUS, 40$                      : 0945
                        0818    C8  D5 0051E          TSTL    FILENAME_WIDTH                   :
                                66  19 00522          BLSS    40$                              :
                                05  12 00524          BNEQ    38$                              : 0951
              0818  C8          13  D0 00526          MOVL    #19, FILENAME_WIDTH              :
                          2C    AE  9F 0052B 38$:     PUSHAB  VALUE_DESC                       : 0952
                        0324    CA  9F 0052E          PUSHAB  P.ADG                            :
         00000000G  00          02  FB 00532          CALLS   #2, CLI$GET_VALUE                :
                        081C    C8  9F 00539          PUSHAB  OWNER_WIDTH                      : 0953
                          34    AE  DD 0053D          PUSHL   VALUE_DESC+4                     : 0954
                  7E      34    AE  3C 00540          MOVZWL  VALUE_DESC, -(SP)                : 0953
         00000000G  00          03  FB 00544          CALLS   #3, LIB$CVT_DTB                  :
                  57            50  D0 0054B          MOVL    R0, STATUS                       :
                  39            57  E9 0054E          BLBC    STATUS, 40$                      : 0956
                        081C    C8  D5 00551          TSTL    OWNER_WIDTH                      :
                                33  19 00555          BLSS    40$                              :
                                05  12 00557          BNEQ    39$                              : 0962
              081C  C8          14  D0 00559          MOVL    #20, OWNER_WIDTH                 :
                          2C    AE  9F 0055E 39$:     PUSHAB  VALUE_DESC                       : 0963
                        0338    CA  9F 00561          PUSHAB  P.ADI                            :
         00000000G  00          02  FB 00565          CALLS   #2, CLI$GET_VALUE                :
                        0820    C8  9F 0056C          PUSHAB  SIZE_WIDTH                       : 0964
                          34    AE  DD 00570          PUSHL   VALUE_DESC+4                     : 0965
                  7E      34    AE  3C 00573          MOVZWL  VALUE_DESC, -(SP)                : 0964
         00000000G  00          03  FB 00577          CALLS   #3, LIB$CVT_DTB                  :
                  57            50  D0 0057E          MOVL    R0, STATUS                       :
                  06            57  E9 00581          BLBC    STATUS, 40$                      : 0967
                        0820    C8  D5 00584          TSTL    SIZE_WIDTH                       :
                                3B  18 00588          BGEQ    42$                              :
                        0830    C8  9F 0058A 40$:     PUSHAB  OUTPUT_RAB                       : 0970
         00000000G  00          01  FB 0058E          CALLS   #1, SYS$FLUSH                    :
                        0830    C8  9F 00595          PUSHAB  OUTPUT_RAB                       :
         00000000G  00          01  FB 00599          CALLS   #1, SYS$WAIT                     :
                          2C    AE  9F 005A0          PUSHAB  VALUE_DESC                       :
                                01  DD 005A3          PUSHL   #1                               :
                    007910FC    8F  DD 005A5          PUSHL   #7934204                         :
```

```
                    000000000G  00         03  FB  005AB        CALLS   #3, LIB$SIGNAL                          :
     04       14  AB            03         00  ED  005B2        CMPZV   #0, #3, WORST_ERROR, #4                 :
                                           08  18  005B8        BGEQ    41$                                     :
                      14        AB 107910FC  8F  D0  005BA        MOVL    #276369660, WORST_ERROR                 :
                                         038C  31  005C2  41$:  BRW     86$                                     : 0971
                                           05  12  005C5  42$:  BNEQ    43$                                     : 0973
                    0820        C8         06  D0  005C7        MOVL    #6, SIZE_WIDTH                          :
                                           5B  DD  005CC  43$:  PUSHL   R11                                     : 0978
                    000000000G  00         01  FB  005CE        CALLS   #1, SYS$CREATE                          :
                                57         50  D0  005D5        MOVL    R0, STATUS                              :
                                11         57  E9  005D8        BLBC    STATUS, 44$                             : 0979
                                      0830  C8  9F  005DB        PUSHAB  OUTPUT_RAB                              : 0985
                    000000000G  00         01  FB  005DF        CALLS   #1, SYS$CONNECT                         :
                                57         50  D0  005E6        MOVL    R0, STATUS                              :
                                0B         57  E8  005E9        BLBS    STATUS, 45$                             : 0986
                                           5B  DD  005EC  44$:  PUSHL   R11                                     : 0989
                                  007910A4  8F  DD  005EE        PUSHL   #7934116                                :
                                         0355  31  005F4        BRW     85$                                     :
                3E        40    AB         02  E1  005F7  45$:  BBC     #2, OUTPUT_FAB+64, 46$                  : 0995
     1C         00              6E         00  2C  005FC        MOVC5   #0, (SP), #0, #28, GETDVI_ARGS          : 0998
                                        10  AE      00601                                                       :
                      10        AE 00040004  8F  D0  00603        MOVL    #262148, GETDVI_ARGS                    : 0999
                      14        AE        04  AE  9E  0060B        MOVAB   INDEV_CLASS, GETDVI_ARGS+4              : 1000
                      1C        AE 00080004  8F  D0  00610        MOVL    #524292, GETDVI_ARGS+12                 : 1001
                      20        AE        08  AE  9E  00618        MOVAB   INDEV_BUFSIZ, GETDVI_ARGS+16            : 1002
                                           7E  7C  0061D        CLRQ    -(SP)                                   : 1005
                                           7E  7C  0061F        CLRQ    -(SP)                                   :
                                        20  AE  9F  00621        PUSHAB  GETDVI_ARGS                             :
                                      034C  CA  9F  00624        PUSHAB  P.ADK                                   :
                                           7E  7C  00628        CLRQ    -(SP)                                   :
                    000000000G  00         08  FB  0062A        CALLS   #8, SYS$GETDVI                          :
                                57         50  D0  00631        MOVL    R0, STATUS                              :
                                03         57  E8  00634        BLBS    STATUS, 46$                             : 1006
                                         01CA  31  00637        BRW     71$                                     :
                                      0814  C8  D5  0063A  46$:  TSTL    DISPLAY_WIDTH                           : 1013
                                           15  12  0063E        BNEQ    48$                                     :
                    00000042    8F        04  AE  D1  00640        CMPL    INDEV_CLASS, #66                        : 1016
                                           05  13  00648        BEQL    47$                                     :
                      08        AE        84  8F  9A  0064A        MOVZBL  #132, INDEV_BUFSIZ                      :
                    0814        C8        08  AE  D0  0064F  47$:  MOVL    INDEV_BUFSIZ, DISPLAY_WIDTH             : 1017
                18              68         03  E0  00655  48$:  BBS     #3, QUAL_FLAGS, 49$                     : 1023
                13        01    A8         05  E0  00659        BBS     #5, QUAL_FLAGS+1, 49$                   :
                                        01  A8  95  0065E        TSTB    QUAL_FLAGS+1                            : 1024
                                           0E  19  00661        BLSS    49$                                     :
                09        02    A8         03  E0  00663        BBS     #3, QUAL_FLAGS+2, 49$                   :
                                05        01  A8  E8  00668        BLBS    QUAL_FLAGS+1, 49$                       : 1025
                09        01    A8         03  E0  0066C        BBS     #3, QUAL_FLAGS+1, 50$                   :
                                        03  A8  95  00671  49$:  TSTB    QUAL_FLAGS+3                            : 1026
                                           04  18  00674        BGEQ    50$                                     :
                      08        A8         01  D0  00676        MOVL    #1, COLUMN_COUNT                        : 1027
                                1D        04  A8  E9  0067A  50$:  BLBC    QUAL_FLAGS+4, 52$                       : 1031
                                      0668  C8  D5  0067E        TSTL    FIRST_XAB                               : 1034
                                           0C  12  00682        BNEQ    51$                                     :
                                56      0738  C8  9E  00684        MOVAB   INFO_XABFHC, XAB_PTR                    : 1035
                    0668        C8         56  D0  00689        MOVL    XAB_PTR, FIRST_XAB                      :
                                           0B  11  0068E        BRB     52$                                     :
                      04        A6      0738  C8  9E  00690  51$:  MOVAB   INFO_XABFHC, 4(XAB_PTR)                 : 1036
```

```
                                                          I 16
DIRECTORY                                          15-Sep-1984 23:38:58    VAX-11 Bliss-32 V4.0-742              Page  31
V04-000                                            14-Sep-1984 12:19:31    [DIR.SRC]DIRECTORY.B32;1                     (4)

                    56    0738  C8  9E 00696        MOVAB   INFO_XABFHC, XAB_PTR                          :
      1D     04     A8          01  E1 0069B 52$:   BBC     #1, QUAL_FLAGS+4, 54$                         : 1038
                          0668  C8  D5 006A0        TSTL    FIRST_XAB                                     : 1041
                                0C  12 006A4        BNEQ    53$                                           :
                    56    070C  C8  9E 006A6        MOVAB   INFO_XABDAT, XAB_PTR                          : 1042
           0668     C8          56  D0 006AB        MOVL    XAB_PTR, FIRST_XAB                            :
                                0B  11 006B0        BRB     54$                                           :
             04     A6    070C  C8  9E 006B2 53$:   MOVAB   INFO_XABDAT, 4(XAB_PTR)                       : 1043
                    56    070C  C8  9E 006B8        MOVAB   INFO_XABDAT, XAB_PTR                          :
      1D     04     A8          02  E1 006BD 54$:   BBC     #2, QUAL_FLAGS+4, 56$                         : 1045
                          0668  C8  D5 006C2        TSTL    FIRST_XAB                                     : 1048
                                0C  12 006C6        BNEQ    55$                                           :
                    56    06B4  C8  9E 006C8        MOVAB   INFO_XABPRO, XAB_PTR                          : 1049
           0668     C8          56  D0 006CD        MOVL    XAB_PTR, FIRST_XAB                            :
                                0B  11 006D2        BRB     56$                                           :
             04     A6    06B4  C8  9E 006D4 55$:   MOVAB   INFO_XABPRO, 4(XAB_PTR)                       : 1050
                    56    06B4  C8  9E 006DA        MOVAB   INFO_XABPRO, XAB_PTR                          :
      1D     04     A8          03  E1 006DF 56$:   BBC     #3, QUAL_FLAGS+4, 58$                         : 1052
                          0668  C8  D5 006E4        TSTL    FIRST_XAB                                     : 1055
                                0C  12 006E8        BNEQ    57$                                           :
                    56    06A8  C8  9E 006EA        MOVAB   INFO_XABSUM, XAB_PTR                          : 1056
           0668     C8          56  D0 006EF        MOVL    XAB_PTR, FIRST_XAB                            :
                                0B  11 006F4        BRB     58$                                           :
             04     A6    06A8  C8  9E 006F6 57$:   MOVAB   INFO_XABSUM, 4(XAB_PTR)                       : 1057
                    56    06A8  C8  9E 006FC        MOVAB   INFO_XABSUM, XAB_PTR                          :
      45     04     A8          04  E1 00701 58$:   BBC     #4, QUAL_FLAGS+4, 61$                         : 1059
                          0668  C8  D5 00706        TSTL    FIRST_XAB                                     : 1062
                                0C  12 0070A        BNEQ    59$                                           :
                    56    066C  C8  9E 0070C        MOVAB   INFO_XABJNL, XAB_PTR                          : 1063
           0668     C8          56  D0 00711        MOVL    XAB_PTR, FIRST_XAB                            :
                                0B  11 00716        BRB     60$                                           :
             04     A6    066C  C8  9E 00718 59$:   MOVAB   INFO_XABJNL, 4(XAB_PTR)                       : 1064
                    56    066C  C8  9E 0071E        MOVAB   INFO_XABJNL, XAB_PTR                          :
                    50      1C  A8  D0 00723 60$:   MOVL    DISPLAY_BLOCK, R0                             : 1065
           0684     C8    0199  C0  9E 00727        MOVAB   409(R0), INFO_XABJNL+24                       :
           0680     C8          10  90 0072E        MOVB    #16, INFO_XABJNL+20                           : 1066
           067C     C8    01AA  C0  9E 00733        MOVAB   426(R0), INFO_XABJNL+16                       : 1067
           0678     C8          10  90 0073A        MOVB    #16, INFO_XABJNL+12                           : 1068
           068C     C8    01BB  C0  9E 0073F        MOVAB   443(R0), INFO_XABJNL+32                       : 1069
           0688     C8          10  90 00746        MOVB    #16, INFO_XABJNL+28                           : 1070
      50     10     A8    0818  C8  C1 0074B 61$:   ADDL3   FILENAME_WIDTH, COLUMN_WIDTH, R0              : 1078
             10     A8      01  A0  9E 00752        MOVAB   1(R0), COLUMN_WIDTH                           :
      0C     01     A8          05  E1 00757        BBC     #5, QUAL_FLAGS+1, 62$                         : 1079
      50     10     A8    081C  C8  C1 0075C        ADDL3   OWNER_WIDTH, COLUMN_WIDTH, R0                 :
             10     A8      02  A0  9E 00763        MOVAB   2(R0), COLUMN_WIDTH                           :
      22     02     A8          03  E1 00768 62$:   BBC     #3, QUAL_FLAGS+2, 64$                         : 1080
      11     02     A8          04  E1 0076D        BBC     #4, QUAL_FLAGS+2, 63$                         : 1083
                    50    0820  C8  D0 00772        MOVL    SIZE_WIDTH, R0                                : 1084
             10     A8      10 B840  3E 00777        MOVAW   @COLUMN_WIDTH[R0], COLUMN_WIDTH               :
             10     A8          02  C0 0077D        ADDL2   #2, COLUMN_WIDTH                              :
                                0C  11 00781        BRB     64$                                           :
      50     10     A8    0820  C8  C1 00783 63$:   ADDL3   SIZE_WIDTH, COLUMN_WIDTH, R0                  : 1085
             10     A8      02  A0  9E 0078A        MOVAB   2(R0), COLUMN_WIDTH                           :
      1F            68          04  E0 0078F 64$:   BBS     #4, QUAL_FLAGS, 65$                           : 1087
      1B            68          06  E0 00793        BBS     #6, QUAL_FLAGS, 65$                           :
      17            68          05  E0 00797        BBS     #5, QUAL_FLAGS, 65$                           : 1088
                                68  95 0079B        TSTB    QUAL_FLAGS                                    :
```

```
                                               13 19  0079D        BLSS    65$
                0E           01 A8             05 E0  0079F        BBS     #5, QUAL_FLAGS+1, 65$                      1089
                                          01   A8 95  007A4        TSTB    QUAL_FLAGS+1
                                               09 19  007A7        BLSS    65$
                04           02 A8             03 E0  007A9        BBS     #3, QUAL_FLAGS+2, 65$                      1090
                                19        01   A8 E9  007AE        BLBC    QUAL_FLAGS+1, 66$
                             10 A8             04 C0  007B2  65$:  ADDL2   #4, COLUMN_WIDTH                           1093
                51         0814 C8             04 C1  007B6        ADDL3   #4, DISPLAY_WIDTH, R1                      1094
                                51        10   A8 C6  007BC        DIVL2   COLUMN_WIDTH, R1
                                50        08   A8 D0  007C0        MOVL    COLUMN_COUNT, R0
                                51             50 D1  007C4        CMPL    R0, R1
                                               12 1A  007C7        BGTRU   67$
                                               13 11  007C9        BRB     68$
                51         0814 C8        10   A8 C7  007CB  66$:  DIVL3   COLUMN_WIDTH, DISPLAY_WIDTH, R1            1096
                                50        08   A8 D0  007D2        MOVL    COLUMN_COUNT, R0
                                51             50 D1  007D6        CMPL    R0, R1
                                               03 1B  007D9        BLEQU   68$
                                50             51 D0  007DB  67$:  MOVL    R1, R0
                             08 A8             50 D0  007DE  68$:  MOVL    R0, COLUMN_COUNT
                                               03 15  007E2        BLEQ    69$                                        1097
                                04             68 E9  007E4        BLBC    QUAL_FLAGS, 70$
                             08 A8             01 D0  007E7  69$:  MOVL    #1, COLUMN_COUNT
                                          18   A8 9F  007EB  70$:  PUSHAB  CMN_QUAL_CTX                               1102
                             04 AE      01FE   8F 3C  007EE        MOVZWL  #510, 4(SP)                                1108
                                          04   AE 9F  007F4        PUSHAB  4(SP)                                      1102
                      00000000G 00             02 FB  007F7        CALLS   #2, LIB$QUAL_FILE_PARSE
                                57             50 D0  007FE        MOVL    R0, STATUS
                                37             57 E8  00801        BLBS    STATUS, 74$                                1111
                                        0830   C8 9F  00804  71$:  PUSHAB  OUTPUT_RAB                                 1114
                      00000000G 00             01 FB  00808        CALLS   #1, SYS$FLUSH
                                        0830   C8 9F  0080F        PUSHAB  OUTPUT_RAB
                      00000000G 00             01 FB  00813        CALLS   #1, SYS$WAIT
                                               57 DD  0081A        PUSHL   STATUS
                      00000000G 00             01 FB  0081C        CALLS   #1, LIB$SIGNAL
                                07             57 93  00823        BITB    STATUS, #7
                                               03 12  00826        BNEQ    73$
                                             0126 31  00828  72$:  BRW     86$
50              57              03             00 EF  0082B  73$:  EXTZV   #0, #3, STATUS, R0
50        14    A8              03             00 ED  00830        CMPZV   #0, #3, WORST_ERROR, R0
                                               F0 18  00836        BGEQ    72$
                                             F88B 31  00838        BRW     2$
                                          34   AE 9F  0083B  74$:  PUSHAB  FILE_DESC                                  1118
                                        035C   CA 9F  0083E        PUSHAB  P.ADM
                      00000000G 00             02 FB  00842        CALLS   #2, CLI$GET_VALUE
                             DC AD        38   AE D0  00849        MOVL    FILE_DESC+4, INPUT_FAB+44                  1119
                             E4 AD        34   AE 90  0084E        MOVB    FILE_DESC, INPUT_FAB+52                    1120
                03           01 A8             01 E0  00853        BBS     #1, QUAL_FLAGS+1, 75$                      1126
                                1B             68 E9  00858        BLBC    QUAL_FLAGS, 76$                            1127
                                          FC   AB 9F  0085B  75$:  PUSHAB  FORMAT_ACL_ADDR                            1129
                                        0388   CA 9F  0085E        PUSHAB  P.ADQ                                      1130
                                        0370   CA 9F  00862        PUSHAB  P.ADO                                      1129
                      00000000G 00             03 FB  00866        CALLS   #3, LIB$FIND_IMAGE_SYMBOL
                                57             50 D0  0086D        MOVL    R0, STATUS
                                03             57 E8  00870        BLBS    STATUS, 75$                                1131
                                             F81F 31  00873        BRW     1$
                                14        04   A8 E8  00876  76$:  BLBS    QUAL_FLAGS+4, 77$                          1147
                0F           04 A8             01 E0  0087A        BBS     #1, QUAL_FLAGS+4, 77$
```

```
DIRECTORY                                                   K 16
V04-000                                            15-Sep-1984 23:38:58    VAX-11 Bliss-32 V4.0-742                 Page  33
                                                   14-Sep-1984 12:19:31    [DIR.SRC]DIRECTORY.B32;1                       (4)

                       0A       04 A8           02 E0 0087F         BBS     #2, QUAL_FLAGS+4, 77$                                  ; 1148
                       05       04 A8           03 E0 00884         BBS     #3, QUAL_FLAGS+4, 77$                                  ;
                       0C       04 A8           04 E1 00889         BBC     #4, QUAL_FLAGS+4, 78$                                  ; 1149
                              FF58 CD        40 8F 88 0088E  77$:   BISB2   #64, INPUT_NAM+8                                       ; 1152
                                D4 AD      0668 C8 D0 00894         MOVL    FIRST_XAB, INPUT_FAB+36                                ; 1153
                                             0C AE 9F 0089A  78$:   PUSHAB  SCAN_CONTEXT                                           ; 1156
                                          0000V CF 9F 0089D         PUSHAB  DIR$INPUT_ERROR                                        ;
                                          0000G CF 9F 008A1         PUSHAB  DIR$GET_INFO                                           ;
                                             B0 AD 9F 008A5         PUSHAB  INPUT_FAB                                              ;
                         00000000G 00           04 FB 008A8         CALLS   #4, LIB$FILE_SCAN                                      ;
                                             B0 AD 9F 008AF         PUSHAB  INPUT_FAB                                              ; 1161
                             0000V CF           01 FB 008B2         CALLS   #1, DIR$GET_FILE                                       ;
                                   BC           50 E8 008B7         BLBS    R0, 76$                                                ;
                                             34 A8 B5 008BA         TSTW    LINE_DESC                                              ; 1163
                                                0A 13 008BD         BEQL    79$                                                    ;
                                             34 A8 9F 008BF         PUSHAB  LINE_DESC                                              ;
                                                7E D4 008C2         CLRL    -(SP)                                                  ;
                             0000V CF           02 FB 008C4         CALLS   #2, DIR$OUTPUT                                         ;
                                           0444 C8 D5 008C9  79$:   TSTL    TOTAL_FILES                                            ; 1164
                                                05 13 008CD         BEQL    80$                                                    ;
                             0000G CF           00 FB 008CF         CALLS   #0, DIR$TOTAL                                          ;
                                   01      0454 C8 D1 008D4  80$:   CMPL    GRAND_DIRS, #1                                         ; 1165
                                                05 14 008D9         BGTR    81$                                                    ;
                       05       01 A8           02 E1 008DB         BBC     #2, QUAL_FLAGS+1, 82$                                  ; 1166
                             0000G CF           00 FB 008E0  81$:   CALLS   #0, DIR$GRAND_TOTAL                                    ; 1167
                                   4C        14 A8 E9 008E5  82$:   BLBC    WORST_ERROR, 84$                                       ; 1172
                       47       04 A8           05 E0 008E9         BBS     #5, QUAL_FLAGS+4, 84$                                  ;
                                           0830 C8 9F 008EE         PUSHAB  OUTPUT_RAB                                             ; 1175
                         00000000G 00           01 FB 008F2         CALLS   #1, SYS$FLUSH                                          ;
                                           0830 C8 9F 008F9         PUSHAB  OUTPUT_RAB                                             ;
                         00000000G 00           01 FB 008FD         CALLS   #1, SYS$WAIT                                           ;
                                      00000000G 8F DD 00904         PUSHL   #DIR$_NOFILES                                          ;
                         00000000G 00           01 FB 0090A         CALLS   #1, LIB$SIGNAL                                         ;
                                      00000000* 8F D5 00911         TSTL    #<DIR$_NOFILES&7>                                      ;
                                                14 13 00917         BEQL    83$                                                    ;
00000000* 8F      14   A8          03           00 ED 00919         CMPZV   #0, #3, WORST_ERROR, #<DIR$_NOFILES&7>                 ;
                                                08 18 00923         BGEQ    83$                                                    ;
                                14 A8 00000000* 8F D0 00925         MOVL    #<DIR$_NOFILES!268435456>, WORST_ERROR                 ;
                                14 A8  10018290 8F D0 0092D  83$:   MOVL    #268534416, WORST_ERROR                                ; 1177
                                                5B DD 00935  84$:   PUSHL   R11                                                    ; 1180
                         00000000G 00           01 FB 00937         CALLS   #1, SYS$CLOSE                                          ;
                                   57           50 D0 0093E         MOVL    R0, STATUS                                             ;
                                   0D           57 E8 00941         BLBS    STATUS, 86$                                            ; 1181
                                                5B DD 00944         PUSHL   R11                                                    ;
                                       0079105A 8F DD 00946         PUSHL   #7934042                                               ;
                             0000V CF           02 FB 0094C  85$:   CALLS   #2, DIR$FILE_ERROR                                     ;
                                   50        14 A8 D0 00951  86$:   MOVL    WORST_ERROR, R0                                        ; 1183
                                                   04 00955         RET                                                            ; 1185

; Routine Size:  2390 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE DIR$GET_FILE (FILE_FAB) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine gets the next file specification in the command line.
1 !       If there are no more files, the routine returns zero.  Otherwise,
1 !       the file specification is placed in the specified FAB for later
1 !       parsing and searching.
1 !
1 ! CALLING SEQUENCE:
1 !       DIR$GET_FILE (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of the FAB into which the file spec is placed
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       1 if a file specification was found
1 !       0 otherwise
1 !
1 ! SIDE EFFECTS:
1 !       The retrieved file specification is placed into the specified
1 !       FAB for later parsing.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FILE_FAB        : REF $BBLOCK;                  ! FAB address
2
2 LOCAL
2       FILE_DESC       : $BBLOCK [DSC$C_S_BLN],        ! File name descr
2       SCAN_FLAGS      : $BBLOCK [4];                  ! $FILESCAN flags
2
2 ! Initialise needed variables.
2
2 CH$FILL (0, DSC$C_S_BLN, FILE_DESC);
2 FILE_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
2
2 ! If there are no more file specifications, return with zero.
2
2 IF NOT CLI$GET_VALUE ($DESCRIPTOR ('INPUT'), FILE_DESC) THEN RETURN 0;
2
2 ! Otherwise, fill in the appropriate fields in the FAB.
2
2 FILE_FAB[FAB$L_FNA] = .FILE_DESC[DSC$A_POINTER];
```

```
;   846         1243  2 FILE_FAB[FAB$B_FNS] = .FILE_DESC[DSC$W_LENGTH];
;   847         1244  2
;   848         1245  2 ! Determine whether or not the new spec is to get a new heading.
;   849         1246  2
;   850         1247  2 SCAN_FLAGS = 0;
;   851         1248  2 $FILESCAN (SRCSTR = FILE_DESC, FLDFLAGS = SCAN_FLAGS);
;   852         1249  2 IF .SCAN_FLAGS[FSCN$V_NODE] OR .SCAN_FLAGS[FSCN$V_DEVICE]
;   853         1250  2 OR .SCAN_FLAGS[FSCN$V_ROOT] OR .SCAN_FLAGS[FSCN$V_DIRECTORY]
;   854         1251  2 THEN
;   855         1252  3     BEGIN
;   856         1253  3     VERSION_INDEX = 0;
;   857         1254  3     PREV_DIR_LEN = PREV_FILE_LEN = 0;
;   858         1255  3     END;
;   859         1256  2
;   860         1257  2 RETURN 1;
;   861         1258  2
;   862         1259  1 END;                                        ! End of routine DIR$GET_FILE


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                             54  55  50  4E  49  00390 P.ADT:   .ASCII  \INPUT\                                          ;
                                                 00395          .BLKB   3
                                       00000005  00398 P.ADS:   .LONG   5                                                ;
                                       00000000' 0039C          .ADDRESS P.ADT                                           ;

                                                                .EXTRN  SYS$FILESCAN

                                                                .PSECT  $CODE$,NOWRT,2

                                             007C 00000 DIR$GET_FILE:
                                                                .WORD   Save R2,R3,R4,R5,R6                              ; 1186
                             56 00000000'   EF    9E 00002      MOVAB   VERSION_INDEX, R6                                ;
                             5E             0C    C2 00009      SUBL2   #12, SP                                          ;
        08          00       6E             00    2C 0000C      MOVC5   #0, (SP), #0, #8, FILE_DESC                      ; 1233
                                       04   AE       00011                                                               ;
                        07   AE             02    90 00013      MOVB    #2, FILE_DESC+3                                  ; 1234
                                       04   AE    9F 00017      PUSHAB  FILE_DESC                                        ; 1238
                                    0000'   CF    9F 0001A      PUSHAB  P.ADS                                            ;
                 00000000G   00             02    FB 0001E      CALLS   #2, CLI$GET_VALUE                                ;
                             3A             50    E9 00025      BLBC    R0, 3$                                           ;
                             50        04   AC    D0 00028      MOVL    FILE_FAB, R0                                     ; 1242
                        2C   A0        08   AE    D0 0002C      MOVL    FILE_DESC+4, 44(R0)                              ;
                        34   A0        04   AE    90 00031      MOVB    FILE_DESC, 52(R0)                                ; 1243
                                            6E    D4 00036      CLRL    SCAN_FLAGS                                       ; 1247
                                            5E    DD 00038      PUSHL   SP                                               ; 1248
                                            7E    D4 0003A      CLRL    -(SP)                                            ;
                                       0C   AE    9F 0003C      PUSHAB  FILE_DESC                                        ;
                 00000000G   00             03    FB 0003F      CALLS   #3, SYS$FILESCAN                                 ;
                             0C             6E    E8 00046      BLBS    SCAN_FLAGS, 1$                                   ; 1249
                  C8         6E             01    E0 00049      BBS     #1, SCAN_FLAGS, 1$                               ;
                  04         6E             02    E0 0004D      BBS     #2, SCAN_FLAGS, 1$                               ; 1250
                  09         6E             03    E1 00051      BBC     #3, SCAN_FLAGS, 2$                               ;
                                            66    D4 00055 1$:  CLRL    VERSION_INDEX                                    ; 1253
                                       F8   A6    D4 00057      CLRL    PREV_FILE_LEN                                    ; 1254
                                     FEF4   C6    D4 0005A      CLRL    PREV_DIR_LEN                                     ;
```

```
                          50         01  D0 0005E 2$:      MOVL    #1, R0                                      ; 1257
                                         04 00061          RET                                                 ;
                                     50  D4 00062 3$:      CLRL    R0                                          ; 1259
                                         04 00064          RET                                                 ;

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 0956
;  864       1260  1 GLOBAL ROUTINE DIR$INPUT_ERROR (FILE_FAB) =
;  865       1261  1
;  866       1262  1 !++
;  867       1263  1 !
;  868       1264  1 ! FUNCTIONAL DESCRIPTION:
;  869       1265  1 !
;  870       1266  1 !       This routine is used to signal errors received on the input file.
;  871       1267  1 !
;  872       1268  1 ! CALLING SEQUENCE:
;  873       1269  1 !       DIR$INPUT_ERROR (ARG1)
;  874       1270  1 !
;  875       1271  1 ! INPUT PARAMETERS:
;  876       1272  1 !       ARG1: address of the FAB
;  877       1273  1 !
;  878       1274  1 ! IMPLICIT INPUTS:
;  879       1275  1 !       none
;  880       1276  1 !
;  881       1277  1 ! OUTPUT PARAMETERS:
;  882       1278  1 !       none
;  883       1279  1 !
;  884       1280  1 ! IMPLICIT OUTPUTS:
;  885       1281  1 !       none
;  886       1282  1 !
;  887       1283  1 ! ROUTINE VALUE:
;  888       1284  1 !       1
;  889       1285  1 !
;  890       1286  1 ! SIDE EFFECTS:
;  891       1287  1 !       The error is signaled by placing the appropriate message into
;  892       1288  1 !       the output file.
;  893       1289  1 !
;  894       1290  1 !--
;  895       1291  1
;  896       1292  2 BEGIN
;  897       1293  2
;  898       1294  2 MAP
;  899       1295  2       FILE_FAB        : REF $BBLOCK;          ! FAB address
;  900       1296  2
;  901       1297  2 IF .FILE_FAB[FAB$L_STS] NEQ RMS$_FNF
;  902       1298  2 THEN DIR$FILE_ERROR (DIR$_OPENIN, .FILE_FAB);
;  903       1299  2
;  904       1300  2 RETURN 1;
;  905       1301  2
;  906       1302  1 END;                                    ! End of routine DIR$INPUT_ERROR



                                     0000 00000          .ENTRY  DIR$INPUT_ERROR, Save nothing          ; 1260
                   50         04  AC  D0 00002          MOVL    FILE_FAB, R0                           ; 1297
         00018292  8F         08  A0  D1 00006          CMPL    8(R0), #98962                          ;
                                  0D  13 0000E          BEQL    1$                                     ;
                                  50  DD 00010          PUSHL   R0                                     ; 1298
                        0079109A  8F  DD 00012          PUSHL   #7934106                               ;
            0000V  CF             02  FB 00018          CALLS   #2, DIR$FILE_ERROR                     ;
                   50             01  D0 0001D  1$:     MOVL    #1, R0                                 ; 1300
                                      04 00020          RET                                            ; 1302
```

```
; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 09BB
```

```
1 GLOBAL ROUTINE DIR$FILE_ERROR (ERROR_CODE, FILE_FAB) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is used to signal errors received on files.
1 !
1 ! CALLING SEQUENCE:
1 !       DIR$FILE_ERROR (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: error code
1 !       ARG2: address of the FAB
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FILE_FAB        : REF $BBLOCK;          ! FAB address
2
2 BIND
2       FILE_NAM        = .FILE_FAB[FAB$L_NAM]  : $BBLOCK;      ! NAMe block address
2
2 LOCAL
2       FILE_NAME       : $BBLOCK [DSC$C_S_BLN];        ! Local file name descr
2
2 CH$FILL (0, DSC$C_S_BLN, FILE_NAME);
2 IF .FILE_NAM[NAM$B_RSL] NEQ 0
2 THEN
3     BEGIN
3     FILE_NAME[DSC$W_LENGTH] = .FILE_NAM[NAM$B_RSL];
3     FILE_NAME[DSC$A_POINTER] = .FILE_NAM[NAM$L_RSA];
3     END
2 ELSE IF .FILE_NAM[NAM$B_ESL] NEQ 0
2 THEN
3     BEGIN
3     FILE_NAME[DSC$W_LENGTH] = .FILE_NAM[NAM$B_ESL];
3     FILE_NAME[DSC$A_POINTER] = .FILE_NAM[NAM$L_ESA];
3     END
2 ELSE
```

```
;  965        1360  3     BEGIN
;  966        1361  3     FILE_NAME[DSC$W_LENGTH] = .FILE_FAB[FAB$B_FNS];
;  967        1362  3     FILE_NAME[DSC$A_POINTER] = .FILE_FAB[FAB$L_FNA];
;  968        1363  2     END;
;  969        1364  2
;  970      P 1365  2 SIGNAL (.ERROR_CODE, 1, FILE_NAME, .FILE_FAB[FAB$L_STS],
;  971        1366  2                                  .FILE_FAB[FAB$L_STV]);
;  972        1367  2
;  973        1368  2 IF .WORST_ERROR EQL (.ERROR_CODE OR STS$M_INHIB_MSG)
;  974        1369  2 THEN WORST_ERROR = .FILE_FAB[FAB$L_STS] OR STS$M_INHIB_MSG;
;  975        1370  2
;  976        1371  2 RETURN 1;
;  977        1372  2
;  978        1373  1 END;                                              ! End of routine DIR$FILE_ERROR



                                          01FC 00000        .ENTRY  DIR$FILE_ERROR, Save R2,R3,R4,R5,R6,R7,R8    ; 1303
                         58 00000000'   EF   9E 00002        MOVAB   WORST_ERROR, R8                              ;
                         5E             08   C2 00009        SUBL2   #8, SP                                       ;
                         57        08   AC   D0 0000C        MOVL    FILE_FAB, R7                                 ; 1341
                         56        28   A7   D0 00010        MOVL    40(R7), R6                                   ;
      08          00     6E             00   2C 00014        MOVC5   #0, (SP), #0, #8, FILE_NAME                  ; 1346
                                             6E 00019                                                             ;
                                   03   A6   95 0001A        TSTB    3(R6)                                        ; 1347
                                        0B   13 0001D        BEQL    1$                                           ;
                         6E        03   A6   9B 0001F        MOVZBW  3(R6), FILE_NAME                             ; 1350
                   04    AE        04   A6   D0 00023        MOVL    4(R6), FILE_NAME+4                           ; 1351
                                        19   11 00028        BRB     3$                                           ; 1347
                                   0B   A6   95 0002A 1$:    TSTB    11(R6)                                       ; 1353
                                        0B   13 0002D        BEQL    2$                                           ;
                         6E        0B   A6   9B 0002F        MOVZBW  11(R6), FILE_NAME                            ; 1356
                   04    AE        0C   A6   D0 00033        MOVL    12(R6), FILE_NAME+4                          ; 1357
                                        09   11 00038        BRB     3$                                           ; 1353
                         6E        34   A7   9B 0003A 2$:    MOVZBW  52(R7), FILE_NAME                            ; 1361
                   04    AE        2C   A7   D0 0003E        MOVL    44(R7), FILE_NAME+4                          ; 1362
                                 081C   C8   9F 00043 3$:    PUSHAB  OUTPUT_RAB                                   ; 1366
               00000000G 00             01   FB 00047        CALLS   #1, SYS$FLUSH                                ;
                                 081C   C8   9F 0004E        PUSHAB  OUTPUT_RAB                                   ;
               00000000G 00             01   FB 00052        CALLS   #1, SYS$WAIT                                 ;
                         7E        08   A7   7D 00059        MOVQ    8(R7), -(SP)                                 ;
                                   08   AE   9F 0005D        PUSHAB  FILE_NAME                                    ;
                                        01   DD 00060        PUSHL   #1                                           ;
                         52        04   AC   D0 00062        MOVL    ERROR_CODE, R2                               ;
                                        52   DD 00066        PUSHL   R2                                           ;
               00000000G 00             05   FB 00068        CALLS   #5, LIB$SIGNAL                               ;
                         07             52   93 0006F        BITB    R2, #7                                       ;
                                        14   13 00072        BEQL    4$                                           ;
      50          52     03             00   EF 00074        EXTZV   #0, #3, R2, R0                               ;
      50          68     03             00   ED 00079        CMPZV   #0, #3, WORST_ERROR, R0                      ;
                                        08   18 0007E        BGEQ    4$                                           ;
                  68     52 10000000    8F   C9 00080        BISL3   #268435456, R2, WORST_ERROR                  ;
      52          01     1C             01   F0 00088 4$:    INSV    #1, #28, #1, R2                              ; 1368
                         52             68   D1 0008D        CMPL    WORST_ERROR, R2                              ;
                                        09   12 00090        BNEQ    5$                                           ;
```

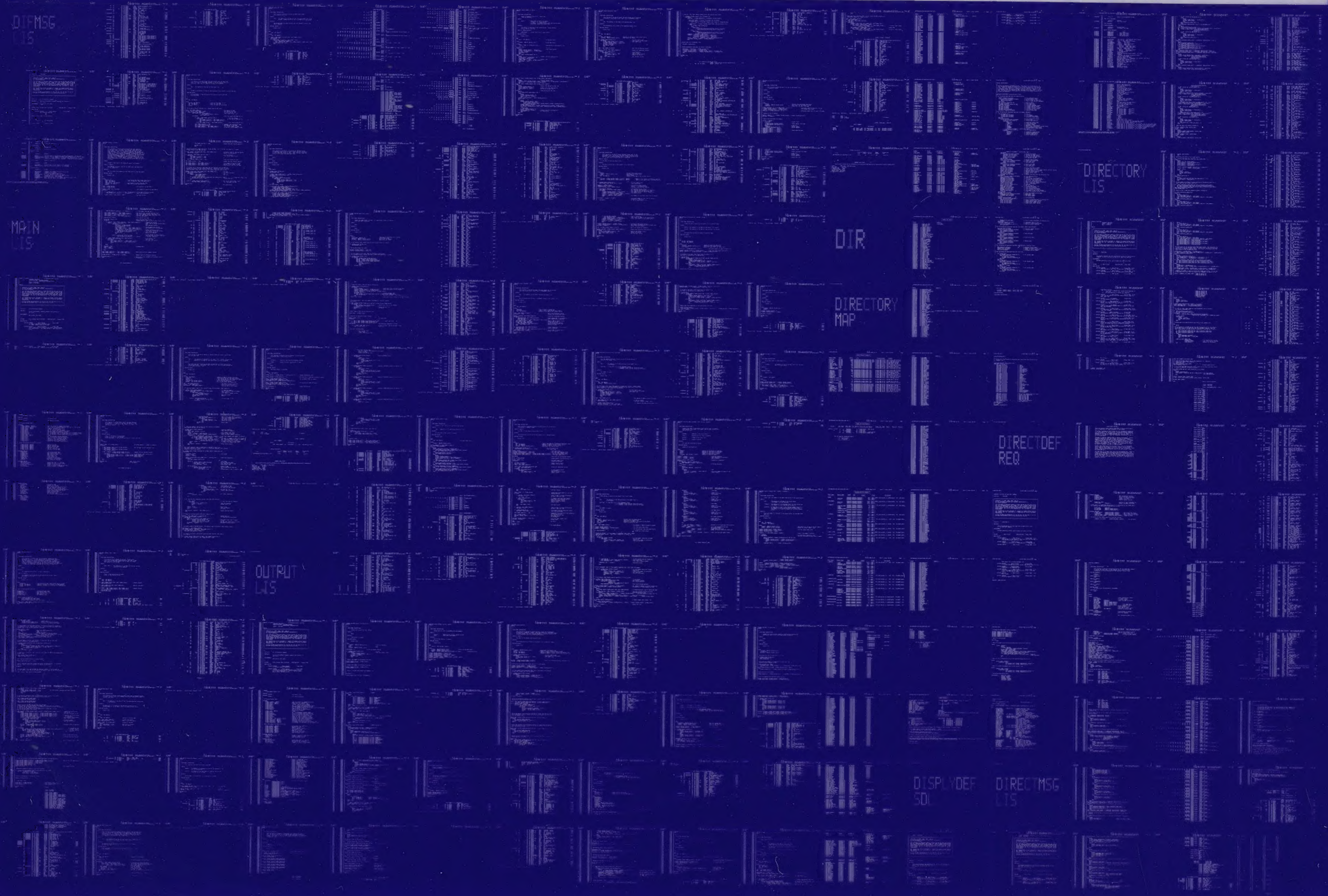
```
                      68      08    A7 10000000    8F   C9 00092          BISL3   #268435456, 8(R7), WORST_ERROR       ; 1369
                                    50             01   D0 0009B 5$:      MOVL    #1, R0                               ; 1371
                                                        04 0009E          RET                                          ; 1373

; Routine Size:  159 bytes,    Routine Base:  $CODE$ + 09DC
```

```
  1 GLOBAL ROUTINE DIR$OUTPUT (MESSAGE_CODE, CONTROL_STRING, ARGS) =
  1
  1 !++
  1 !
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       This routine accepts, as input, an $FAO control string and any
  1 !       arguments to be formatted by the control string.  The formatted
  1 !       line is then written to the desired output file.
  1 !
  1 ! CALLING SEQUENCE:
  1 !       DIR$OUTPUT (ARG1, ARG2, ..., ARGn)
  1 !
  1 ! INPUT PARAMETERS:
  1 !       ARG1: message code for the text to display
  1 !       ARG2: address of the $FAO control string
  1 !       ARG3 - ARGn: arguments to be formatted
  1 !
  1 ! IMPLICIT INPUTS:
  1 !       none
  1 !
  1 ! OUTPUT PARAMETERS:
  1 !       none
  1 !
  1 ! IMPLICIT OUTPTUS:
  1 !       none
  1 !
  1 ! ROUTINE VALUE:
  1 !       1
  1 !
  1 ! SIDE EFFECTS:
  1 !       none
  1 !
  1 !--
  1
  2 BEGIN
  2
  2 MAP
  2         CONTROL_STRIN;  : REF $BBLOCK;          ! Address of the control string
  2
  2 LOCAL
  2         FAO_CTL_STRING  : REF $BBLOCK,          ! Addr of $FAO control string
  2         MESSAGE_DESC    : $BBLOCK [DSC$C_S_BLN],        ! Message text descr
  2         MESSAGE_TEXT    : VECTOR [256, BYTE],   ! Message text
  2         STATUS;                                 ! Local routien exit status
  2
  2 ! If there is a message code present, get the message text via a $GETMSG.
  2 ! Otherwise, use the descriptor supplied.
  2
  2 IF .MESSAGE_CODE NEQ 0
  2 THEN
  3     BEGIN
  3     CH$FILL (0, DSC$C_S_BLN, MESSAGE_DESC);
  3     MESSAGE_DESC[DSC$W_LENGTH] = 256;
  3     MESSAGE_DESC[DSC$A_POINTER] = MESSAGE_TEXT;
P 3     $GETMSG (MSGID = .MESSAGE_CODE,
P 3             MSGLEN = MESSAGE_DESC[DSC$W_LENGTH],
; 1037        P 1431  3           BUFADR = MESSAGE_DESC,
; 1038          1432  3           FLAGS = 1);
; 1039          1433  3   FAO_CTL_STRING = MESSAGE_DESC;
; 1040          1434  3   END
; 1041          1435  2 ELSE FAO_CTL_STRING = .CONTROL_STRING;
; 1042          1436  2
; 1043          1437  2 ! Format the line.
; 1044          1438  2
; 1045          1439  2 IF .FAO_CTL_STRING NEQA LINE_DESC
; 1046          1440  2 THEN
; 1047          1441  3     BEGIN
; 1048          1442  3     CH$FILL (0, DSC$C_S_BLN, LINE_DESC);
; 1049          1443  3     LINE_DESC[DSC$W_LENGTH] = 1024;
; 1050          1444  3     LINE_DESC[DSC$A_POINTER] = LINE_BUFFER;
; 1051          1445  3
; 1052        P 1446  3     $FAOL (CTRSTR = .FAO_CTL_STRING,
; 1053        P 1447  3            OUTLEN = LINE_DESC,
; 1054        P 1448  3            OUTBUF = LINE_DESC,
; 1055          1449  3            PRMLST = ARGS);
; 1056          1450  3
; 1057          1451  3     OUTPUT_RAB[RAB$L_RBF] = .LINE_DESC[DSC$A_POINTER];
; 1058          1452  3     OUTPUT_RAB[RAB$W_RSZ] = .LINE_DESC[DSC$W_LENGTH];
; 1059          1453  3     END
; 1060          1454  2 ELSE
; 1061          1455  3     BEGIN
; 1062          1456  3     OUTPUT_RAB[RAB$L_RBF] = .FAO_CTL_STRING[DSC$A_POINTER];
; 1063          1457  3     OUTPUT_RAB[RAB$W_RSZ] = .FAO_CTL_STRING[DSC$W_LENGTH];
; 1064          1458  2     END;
; 1065          1459  2
; 1066          1460  2 STATUS = $RMS_PUT (RAB = OUTPUT_RAB);
; 1067          1461  2 IF NOT .STATUS THEN DIR$FILE_ERROR (DIR$_WRITEERR, OUTPUT_RAB);
; 1068          1462  2
; 1069          1463  2 LINE_DESC[DSC$W_LENGTH] = 0;
; 1070          1464  2
; 1071          1465  2 RETURN 1;
; 1072          1466  2
; 1073          1467  1 END;                                            ! End of routine DIR$OUTPUT



                                                          .EXTRN  SYS$GETMSG, SYS$FAOL
                                                          .EXTRN  SYS$PUT

                                      00FC 00000          .ENTRY  DIR$OUTPUT, Save R2,R3,R4,R5,R6,R7              ; 1374
                   57 00000000'  EF   9E 00002          MOVAB   LINE_DESC, R7                                      ;
                   5E      FEF8  CE   9E 00009          MOVAB   -264(SP), SP                                       ;
                             04  AC   D5 0000E          TSTL    MESSAGE_CODE                                       ; 1423
                                 2A   13 00011          BEQL    1$                                                 ;
 08         00       6E          00   2C 00013          MOVC5   #0, (SP), #0, #8, MESSAGE_DESC                     ; 1426
                             F8  AD      00018                                                                     ;
              F8  AD       0100  8F   B0 0001A          MOVW    #256, MESSAGE_DESC                                 ; 1427
              FC  AD             6E   9E 00020          MOVAB   MESSAGE_TEXT, MESSAGE_DESC+4                       ; 1428
                  7E             01   7D 00024          MOVQ    #1, -(SP)                                          ; 1432
                             F8  AD   9F 00027          PUSHAB  MESSAGE_DESC                                       ;
                             F8  AD   9F 0002A          PUSHAB  MESSAGE_DESC                                       ;
                             04  AC   DD 0002D          PUSHL   MESSAGE_CODE                                       ;
          00000000G  00          05   FB 00030          CALLS   #5, SYS$GETMSG                                     ;
```

```
                              56       F8   AD  9E 00037        MOVAB   MESSAGE_DESC, FAO_CTL_STRING                 ; 1433
                                            04  11 0003B        BRB     2$                                           ; 1423
                              56       08   AC  D0 0003D 1$:    MOVL    CONTROL_STRING, FAO_CTL_STRING               ; 1435
                              50            67  9E 00041 2$:    MOVAB   LINE_DESC, R0                                ; 1439
                              50            56  D1 00044        CMPL    FAO_CTL_STRING, R0                           ;
                                            2D  13 00047        BEQL    3$                                           ;
         08         00        6E            00  2C 00049        MOVC5   #0, (SP), #0, #8, LINE_DESC                  ; 1442
                                            67     0004E                                                             ;
                              67     0400   8F  B0 0004F        MOVW    #1024, LINE_DESC                             ; 1443
                         04   A7       08   A7  9E 00054        MOVAB   LINE_BUFFER, LINE_DESC+4                     ; 1444
                                       0C   AC  9F 00059        PUSHAB  ARGS                                         ; 1449
                                            57  DD 0005C        PUSHL   R7                                           ;
                                     00C0   8F  BB 0005E        PUSHR   #^M<R6,R7>                                   ;
                  00000000G   00            04  FB 00062        CALLS   #4, SYS$FAOL                                 ;
                       0824   C7       04   A7  D0 00069        MOVL    LINE_DESC+4, OUTPUT_RAB+40                   ; 1451
                       081E   C7            67  B0 0006F        MOVW    LINE_DESC, OUTPUT_RAB+34                     ; 1452
                                            0B  11 00074        BRB     4$                                           ; 1439
                       0824   C7       04   A6  D0 00076 3$:    MOVL    4(FAO_CTL_STRING), OUTPUT_RAB+40             ; 1456
                       081E   C7            66  B0 0007C        MOVW    (FAO_CTL_STRING), OUTPUT_RAB+34              ; 1457
                                     07FC   C7  9F 00081 4$:    PUSHAB  OUTPUT_RAB                                   ; 1460
                  00000000G   00            01  FB 00085        CALLS   #1, SYS$PUT                                  ;
                              0F            50  E8 0008C        BLBS    STATUS, 5$                                   ; 1461
                                     07FC   C7  9F 0008F        PUSHAB  OUTPUT_RAB                                   ;
                                 007910D4   8F  DD 00093        PUSHL   #7934164                                     ;
                       FEC3   CF            02  FB 00099        CALLS   #2, DIR$FILE_ERROR                           ;
                                            67  B4 0009E 5$:    CLRW    LINE_DESC                                    ; 1463
                              50            01  D0 000A0        MOVL    #1, R0                                       ; 1465
                                                04 000A3        RET                                                  ; 1467

; Routine Size:  164 bytes,    Routine Base:  $CODE$ + 0A7B
; 1075          1468  1 GLOBAL ROUTINE SYS$FORMAT_ACL =
; 1076          1469  1 !++
; 1077          1470  1 !
; 1078          1471  1 ! FUNCTIONAL DESCRIPTION:
; 1079          1472  1 !
; 1080          1473  1 !       This is a dummy routine to satisfy the global reference of
; 1081          1474  1 !       the $FORMAT_ACL macro.  It simply calls the real service,
; 1082          1475  1 !       which has been dynamically loaded.
; 1083          1476  1 !
; 1084          1477  1 ! CALLING SEQUENCE:
; 1085          1478  1 !       via $FORMAT_ACL macro
; 1086          1479  1 !
; 1087          1480  1 ! INPUT PARAMETERS:
; 1088          1481  1 !
; 1089          1482  1 ! IMPLICIT INPUTS:
; 1090          1483  1 !       FORMAT_ACL_ADDR contains the loaded address of SYS$FORMAT_ACL
; 1091          1484  1 !
; 1092          1485  1 ! OUTPUT PARAMETERS:
; 1093          1486  1 !       none
; 1094          1487  1 !
; 1095          1488  1 ! IMPLICIT OUTPTUS:
; 1096          1489  1 !       none
; 1097          1490  1 !
; 1098          1491  1 ! ROUTINE VALUE:
; 1099          1492  1 !       status returned from sys$format_acl service
; 1100          1493  1 !
; 1101          1494  1 ! SIDE EFFECTS:
; 1102          1495  1 !       none
; 1103          1496  1 !
; 1104          1497  1 !--
; 1105          1498  2 BEGIN
; 1106          1499  2 BUILTIN
; 1107          1500  2     CALLG,AP;
; 1108          1501  2
; 1109          1502  2 LOCAL
; 1110          1503  2     STATUS;
; 1111          1504  2
; 1112          1505  2 RETURN CALLG(.AP,.FORMAT_ACL_ADDR)
; 1113          1506  1 END;



                                   0000 00000        .ENTRY  SYS$FORMAT_ACL, Save nothing           ; 1468
              0000'  DF          6C  FA 00002        CALLG   (AP), @FORMAT_ACL_ADDR                 ; 1505
                                       04 00007        RET                                            ; 1506

; Routine Size:  8 bytes,    Routine Base:  $CODE$ + 0B1F


; 1114          1507  1
; 1115          1508  1 END
; 1116          1509  0 ELUDOM
;                                PSECT SUMMARY
;
;       Name                  Bytes                       Attributes
;
;  DIR$COMMON                    2164  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(0)
;  $OWN$                          691  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         928  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        2855  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                          Library Statistics
;
;                                            -------- Symbols --------       Pages      Processing
;       File                                 Total    Loaded   Percent       Mapped     Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1             18619       190         1       1000         00:01.9





;                                COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DIRECTORY/OBJ=OBJ$:DIRECTORY MSRC$:DIRECTORY/UPDATE=(ENH$:DIRECTORY)

; Size:           2855 code + 3783 data bytes
; Run Time:          00:59.6
; Elapsed Time:      02:56.1
; Lines/CPU Min:       1518
; Lexemes/CPU-Min: 28952
; Memory Used:  746 pages
; Compilation Complete
```

0103 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



DIFMSG
LIS

MAIN
LIS

OUTPUT
LIS

DIRECTORY
LIS

DIR

DIRECTORY
MAP

DIRECTDEF
REQ

DISPLYDEF
SDL

DIRECTMSG
LIS

0104 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

[illegible]

DISMOUNT
MAP

DISKQUOTA
LIS

DISMOU

DISKQ

DISMNTSHR
MAP

DISKQUOTA
MAP

DISPLAY
LIS